S7-300 Instruction List

CPU 312, CPU 314, CPU 315-2 DP, CPU 315-2 PN/DP, CPU 317-2 PN/DP, CPU 319-3 PN/DP, IM151-8 PN/DP CPU, IM 154-8 PN/DP CPU

This instruction list is part of the
documentation package with the order number:

6ES7398-8FA10-8BA0

05/2010

A5E02354744-03

Siemens AG
Industry Sector
Postfach 4848
90437 NÜRNBERG / GERMANY

Disclaim of Liability

We have checked the contents of this manual for agreement with the hardware and software described. Since deviations cannot be precluded entirely, we cannot guarantee full agreement. However, the data in this manual are reviewed regularly and any necessary corrections included in subsequent editions. Suggestions for improvement are welcomed.

A5E02354744-03

Contents

Validity Range of the Instructions List

CPU	As of order no.	As of Version Firmware	In the following referred to as
CPU 312	6ES7 312-1AE14-0AB0	V3.0	312
CPU 314	6ES7 314-1AG14-0AB0	V3.0	314
CPU 315-2 DP	6ES7 315-2AH14-0AB0	V3.0	315
CPU 315-2 PN/DP	6ES7 315-2EH14-0AB0	V3.2.1	315
CPU 317-2 PN/DP	6ES7 317-2EK14-0AB0	V3.2.1	317
CPU 319-3 PN/DP	6ES7 319-3EL01-0AB0	V3.2.1	319
IM151-8 PN/DP CPU	6ES7 151-8AB01-0AB0	V3.2	151
IM154-8 PN/DP CPU	6ES7 154-8AB01-0AB0	V3.2	154

Address Identifiers and Parameter Ranges

Addr. ID	Parameter Ranges					Description
	312	314, 151	315, 154	317	319	
Q	0.0 to 127.7 (can be set up 1023.7)	0.0 to 127.7 (can be set up1023.7)	0.0 to 127.7 (can be set up 2047.7)	0.0 to 255.7 (can be set up 8191.7)	0.0 to 255.7 (can be set up 8191.7)	Output (in PIQ)
QB	0 to 127 (can be set up 1023)	0 to 127 (can be set up 1023)	0 to 127 (can be set up 2047)	0 to 255 (can be set up 8191)	0 to 255 (can be set up 8191)	Output byte (in PIQ)
QW	0 to 126 (can be set up 1022)	0 to126 (can be set up 1022)	0 to 126 (can be set up 2046)	0 to 254 (can be set up 8190)	0 to 254 (can be set up 8190)	Output word (in PIQ)
QD	0 to 124 (can be set up 1020)	0 to 124 (can be set up 1020)	0 to 124 (can be set up 2044)	0 to 252 (can be set up 8188)	0 to 252 (can be set up 8188)	Output double word (in PIQ)
DB	1 to 16000	1 to 16000	1 to 16000	1 to 16000	1 to 16000	Data block
DBX	0.0 to 32731.7	0.0 to 65533.7	0.0 to 65533.7	0.0 to 65533.7	0.0 to 65533.7	Data bit in data block
DBB	0 to 32731	0 to 65533	0 to 65533	0 to 65533	0 to 65533	Data byte in DB
DBW	0 to 32730	0 to 65532	0 to 65532	0 to 65532	0 to 65532	Data word in DB
DBD	0 to 32728	0 to 65530	0 to 65530	0 to 65530	0 to 65530	Data double word in DB
DI	1 to 16000	1 to 16000	1 to 16000	1 to 16000	1 to 16000	Instance data block
DIX	0.0 to 32731.7	0.0 to 65533.7	0.0 to 65533.7	0.0 to 65533.7	0.0 to 65533.7	Data bit in instance DB

Addr. ID	Parameter Ranges					Description
	312	314, 151	315, 154	317	319	
DIB	0 to 32731	0 to 65533	0 to 65533	0 to 65533	0 to 65533	Data byte in instance DB
DIW	0 to 32730	0 to 65532	0 to 65532	0 to 65532	0 to 65532	Data word in instance DB
DID	0 to 32728	0 to 65530	0 to 65530	0 to 65530	0 to 65530	Data double word in instance DB
I	0.0 to 127.7 (can be set up 1023.7)	0.0 to 127.7 (can be set up 1023.7)	0.0 to 127.7 (can be set up 2047.7)	0.0 to 255.7 (can be set up 8191.7)	0.0 to 255.7 (can be set up 8191.7)	Inputs (in PII)
IB	0.0 to 127 (can be set up 1023)	0.0 to 127 (can be set up 1023)	0.0 to 127 (can be set up 2047)	0.0 to 255 (can be set up 8191)	0.0 to 255 (can be set up 8191)	Input byte (in PII)
IW	0.0 to 126 (can be set up 1022)	0.0 to 126 (can be set up 1022)	0.0 to 126 (can be set up 2046)	0.0 to 254 (can be set up 8190)	0.0 to 254 (can be set up 8190)	Input word (in PII)
ID	0.0 to 124 (can be set up 1020)	0.0 to 124 (can be set up 1020)	0.0 to 124 (can be set up 2044)	0.0 to 252 (can be set up 8188)	0.0 to 252 (can be set up 8188)	Input double word (in PII
M	0.0 to 255.7	0.0 to 255.7	0.0 to 2047.7	0.0 to 4095.7	0.0 to 4095.7	Bit memory bit
MB	0.0 to 255	0.0 to 255	0.0 to 2047	0.0 to 4095	0.0 to 4095	Bit memory byte
MW	0.0 to 254	0.0 to 254	0.0 to 2046	0.0 to 4094	0.0 to 4094	Bit memory word
MD	0.0 to 252	0.0 to 252	0.0 to 2044	0.0 to 4092	0.0 to 4092	Bit memory double word

Addr. ID	Parameter Ranges					Description
	312	314, 151	315, 154	317	319	
L [1]	0.0 to 2047.7				0.0 to 2047.7	Local data bit
LB [1]	0.0 to 2047				0.0 to 2047	Local data byte
LW [1]	0.0 to 2046				0.0 to 2046	Local data word
LD [1]	0.0 to 2044				0.0 to 2044	Local data double word
Addr. ID	**Parameter Ranges**				**Parameter Ranges**	**Description**
	312	**314**	**315**	**317**	**317**	
PQB	0.0 to 1023		0.0 to 2047	0.0 to 8191	0.0 to 8191	Peripheral output byte (direct I/O access)
PQW	0.0 to 1022		0.0 to 2046	0.0 to 8190	0.0 to 8190	Peripheral output word (direct I/O access)
PQD	0.0 to 1020		0.0 to 2044	0.0 to 8188	0.0 to 8188	Peripheral output double word (direct I/O access)
PIB	0.0 to 1023		0.0 to 2047	0.0 to 8191	0.0 to 8191	Peripheral input byte (direct I/O access)
PIW	0.0 to 1022		0.0 to 2046	0.0 to 8190	0.0 to 8190	Peripheral input word (direct I/O access)
PID	0.0 to 1020		0.0 to 2044	0.0 to 8188	0.0 to 8188	Peripheral input double word (direct I/O access)
T	0.0 to 255			0.0 to 511	0.0 to 511	Timer
Z	0.0 to 255			0.0 to 511	0.0 to 511	Counter

[1] When using temporary variables please note that these are only valid within the particular block and are available as parent local data of other blocks called in this block. After exit and renewed block call it is not certain that the temporary variables will still include the same values that were present when the block call was closed previously. Temporary variables are initially undefined during the block call and must always be re-initialized each time they are used the first time in the block.

Constants

Constant	Description
Parameter	Operand, addressed via parameter
B#16#	Byte hexadecimal
W#16#	Word hexadecimal
DW#16#	Double word hexadecimal
D#Date	IEC date constant
L#Integer	32-bit-integer constant
P#Bitpointer	Pointer constant
S5T#Time	S5-time constant 1) (16-bit), T#1D_5H_3M_1S_2MS
T#Time	Time constant (16-/32-bit), T#1D_5H_3M_1S_2MS
TOD#Time	IEC time constant, T#1D_5H_3M_1S_2MS
C#Time	Counter constant (BCD coded)
2#n	Binary constant
B (b1.b2)B (b1.b2, b3,b4)	Constant, 2 or 4 byte

1) Serves for loading the S5-Timer

Abbreviations and Mnemonics

The following abbreviations and mnemonics are used in the Instruction List:

Abbre-viations	Description	Example
k8	8-bit constant	32
k16	16-bit constant	631
k32	32-bit constant	1272 5624
i8	8-bit integer	-155
i16	16-bit integer	+6523
i 32	32-bit integer	-2 222 222
m	P#x.y (pointer)	P#240.3
n	Binary constant	1001 1100
p	Hexadecimal constant	EA12
q	Real number (32-bit floating-point number)	12.34567E+5
LABEL	Symbolic jump address (max. 4 characters)	DEST
a	Byte address	2
b	Bit address	x.1
c	Operand range	I, Q, M, L, DBX, DIX

Abbre-viations	Description	Example
f	Timer/Counter No.	5
g	Operand range	IB, QB, PIB, PAB MB, LB, DBB, DIB
h	Operand range	IW, QW, PIW, PAW MW, LW, DBW, DIW
i	Operand range	ID, QD, PID, PAD MD, LD, DBD, DID
r	Block No.	10
AZ	Range of address memory cell	
BF	Range error (invalid range)	

Registers

ACCU1 and ACCU2 (32 Bits)

The accumulators are registers for processing bytes, words or double words. The operands are loaded into the accumulators, where they are logically gated. The result of the logic operation (RLO) is in ACCU1.

Accumulator designations:

ACCU	Bits
ACCUx (x = 1 to 2)	Bits 0 to 31
ACCUx-L	Bits 0 to 15
ACCUx-H	Bits 16 to 31
ACCUx-LL	Bits 0 to 7
ACCUx-LH	Bits 8 to 15
ACCUx-HL	Bits 16 to 23
ACCUx-HH	Bits 24 to 31

Address Registers AR1 and AR2 (32 Bits)

The address registers contain the area-internal or area-crossing addresses for instructions using indirect addressing. The address registers are 32 bits long.

The area-internal and/or area-crossing addresses have the following syntax:

- Area-internal address

  `00000000 00000bbb bbbbbbbb bbbbbxxx`

- Area-crossing address

  `10000yyy` `00000bbb bbbbbbbb bbbbbxxx`

Legend:

	`b`	Byte address
	`x`	Bit number
	`y`	Area identifier (see section “Examples of Addressing”)

Status Word (16 Bits)

The status word bits are evaluated or set by the instructions.

The status word is 16 bits long.

Bit	Assignment	Description
0	$\overline{\text{FC}}$ 1) 2)	First check bit , Bit cannot be written and evaluated in the user program since it is not updated at program runtime
1	RLO	Result of (previous) logic operation
2	STA 1) 2)	Status, Bit cannot be written and evaluated in the user program since it is not updated at program runtime
3	OR 1) 2)	Or, Bit cannot be written and evaluated in the user program since it is not updated at program runtime
4	OS	Stored overflow
5	OV	Overflow
6	CC 0	Condition code
7	CC 1	Condition code
8	BR	Binary result
9 ... 15	Unassigned	-

1) In the U-Stack display, the value ”0” is always output.
2) In the display to STATUS block and breakpoint, the bit is correctly displayed/refreshed.

Address types

		1st access								2nd access							
	Commands	**I**	**Q**	**M**	**P**	**L**	**DB**	**DI**	**V**	**I**	**Q**	**M**	**P**	**L**	**DB**	**DI**	**V**
	U, UN, O, ON, X, XN, =, R, S, FP, FN-																
Direct	c 0.0	-	-	-	-	-	-	-	-	c	c	c	-	c	c	c	-
Memory indirect	c [AC D 0]	-	-	AC	-	AC	AC	AC	-	c	c	c	-	c	c	c	-
Memory indirect via block parameter	[#par]	-	-	-	-	-	-	-	-	c	c	c	RE	RE	c	c	c
Register indirect, area-internal	c[AR1. P#..] c[AR2, P#..]	-	-	-	-	-	-	-	-	c	c	c	-	c	c	c	-
Register indirect, area-crossing	[AR1. P#..] [AR2, P#..]	-	-	-	-	-	-	-	-	c	c	c	RE	c	c	c	c

For definitions of the abbreviations refer to page 9, for examples of the address types refer to page 17
Definition of abbreviatons: c= operand range (bit); AC= range of address memory cell; RE= range error (invalid range)

		1st access								2nd access							
	Commands	**I**	**Q**	**M**	**P**	**L**	**DB**	**DI**	**V**	**I**	**Q**	**M**	**P**	**L**	**DB**	**DI**	**V**
	L, T-																
Direct	cB 0. cW 0. cD 0	-	-	-	-	-	-	-	-	c	c	c	c	c	c	c	-
Memory indirect	cB[AC D 0] cW[AC D 0] cD]AC D 0]	-	-	AC	-	AC	AC	AC	-	c	c	c	c	c	c	c	-
Memory indirect via block parameter	Bpar, Wpar, Dpar	-	-	-	-	-	-	-	-	c	c	c	c	RE	c	c	c
Register indirect, area-internal	cB[AR1. P#..] cW[AR1. P#..] cD[AR1. P#..] cB[AR2, P#..] cW[AR2, P#..] cD[AR2, P#..]	-	-	-	-	-	-	-	-	c	c	c	c	c	c	c	-
rRegister indirect, area-crossing	B[AR1. P#..] W[AR1. P#..] D[AR1. P#..] B[AR2, P#..] W[AR2, P#..] D[AR2, P#..]	-	-	-	-	-	-	-	-	c	c	c	c	c	c	c	c

For definitions of the abbreviations refer to page 9, for examples of the address types refer to page 17
Definition of abbreviatons: c= operand range (bit); AC= range of address memory cell; RE= range error (invalid range)

		1st access							
	Commands	**I**	**Q**	**M**	**P**	**L**	**DB**	**DI**	**V**
	SI, SV, SE, SS, SA, R, F, L, LC, A, AN, O, ON, X, XN -								
Direct	T 0	-	-	-	-	-	-	-	-
Memory indirect	T[AC W 0]	-	-	AC	-	AC	AC	AC	-
Memory indirect via block parameter	#Tpar	-	-	-	-	-	-	-	-
	S, ZV, ZR, R, F, L, LC, A, AN, O, ON, X, XN -								
Direct	Z 0	-	-	-	-	-	-	-	-
Memory indirect	Z[AC W 0]	-	-	AC	-	AC	AC	AC	-
Memory indirect via block parameter	#Zpar	-	-	-	-	-	-	-	-
	UC, CC -								
Direct	FB 0. FC 0	-	-	-	-	-	-	-	-
Memory indirect	FB[AC W 0], FC[AC W 0]	-	-	AC	-	AC	AC	AC	-
Memory indirect via block parameter	#FBpar, #FCpar,	-	-	-	-	-	-	-	-
	AUF -								
Direct	DB 0. DI 0	-	-	-	-	-	-	-	-
Memory indirect	DB[AC W 0], DI[AC W 0]	-	-	AC	-	AC	AC	AC	-
Memory indirect via block parameter	#DBpar, #FCpar [1)]	-	-	-	-	-	-	-	-

[1)] The STL syntax prohibits opening the 2nd data block as block parameter
For definitions of the abbreviations refer to page 9, for examples of the address types refer to page 17
Definition of abbreviatons: c= operand range (bit); AC= range of address memory cell; RE= range error (invalid range)

Examples of Addressing

Addressing Examples	Description
Immediate Addressing	
L +27	Load 16-bit integer constant “27” into ACCU1
L L#-1	Load 32-bit integer constant “-1” into ACCU1
L 2#1010101010101010	Load binary constant into ACCU1
L DW#16#A0F0BCFD	Load hexadecimal constant into ACCU1
L ’END’	Load ASCII character into ACCU1
L T#500 ms	Load time value into ACCU1
L C#100	Load count value into ACCU1
L B#(100.12)	Load 2-byte constant
L B#(100.12,50.8)	Load 4-byte constant
L P#10.0	Load area-internal pointer into ACCU1
L P#E20.6	Load area-crossing pointer into ACCU1
L -2.5	Load real number into ACCU1
L D#1995-01-20	Load date
L TOD#13:20:33.125	Load time of day

Addressing Examples	Description
Direct Addressing	
A I 0.0	ANDing of input bit 0.0
L IB 1	Load input byte 1 into ACCU1
L IW 0	Load input word 0 into ACCU1
L ID 0	Load input double word 0 into ACCU1
Indirect Addressing of Timers/Counters	
SP T [LW 8]	Start timer; the timer number is in local word 8
CU C [LW 10]	Start counter; the counter number is in local data word 10
Area-Internal Memory-Indirect Addressing	
A I [LD 12] Example: L P#22.2 T LD 12 A I [LD 12]	AND operation: The address of the input is in local data double word 12 as pointer
A I [DBD 1]	AND operation: The address of the input is in data double word 1 of the DB as pointer
A Q [DID 12]	AND operation: The address of the output is in data double word 12 of the instance DB as pointer
A Q [MD 12]	AND operation: The address of the output is in memory LABELr double word 12 of the instance DB as pointer

Addressing Examples	Description
Area-Internal Register-Indirect Addressing	
A I [AR1.P#12.2]	AND operation: The address of the input is calculated from the “pointer value in AR1+ P#12.2”
Area-Crossing Register-Indirect Addressing	

For area-crossing register-indirect addressing, bits 24 to 26 of the address must also contain an area identifier. The address is in the address register.

Area identifier	Coding (binary)	Coding (hex.)	Area
P	1000 0**000**	80	I/O area
I	1000 0**001**	81	Input area
Q	1000 0**010**	82	Output area
M	1000 0**011**	83	Bit memory area
DB	1000 0**100**	84	Data area
DI	1000 0**101**	85	Instance data area
L	1000 0**110**	86	Local data area
VL	1000 0**111**	87	Predecessor local data (access to local data of invoking block)

L B [AR1.P#8.0]	Load byte into ACCU1: The address is calculated from the “pointer value in AR1+ P#8.0”
A [AR1.P#32.3]	AND operation: The address of the operand is calculated from the “pointer value in AR1+ P#32.3”
Addressing Via Parameters	
A Parameter	Addressing via parameters

Examples of how to calculate the pointer

- **Example for sum of bit addresses ≦7:**

  LAR1 P#8.2
  A I [AR1.P#10.2]

  Result: Input 18.4 is addressed (by adding the byte and bit addresses)

- **Example for sum of bit addresses >7:**

  L MD 0 Random pointer, e.g. P#10.5
  LAR1
  A I [AR1.P#10.7]

  Result: Input 21.4 is addressed (by adding the byte and bit addresses with carry)

List of Instructions

This chapter contains the complete list of S7-300 instructions. The descriptions have been kept as concise as possible. You will find a detailed functional description in the various STEP 7 reference manuals.

Note: For indirect addressing and special operands (see page 17 for examples, see page 14 for address types), you have to also add to the execution time a time for the loading of the address or the respective operand (see page 73).

Bit Logic Instructions

Examining the signal state of the addressed instruction and gating the result with the RLO according to the appropriate logic function.

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs				
				312	314, 151	315, 154	317	319
A	1)	AND	1/2	0.10	0.06	0.05	0.03	0.004
AN	1)	AND- NOT						

Status word for: **A, AN**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:	-	-	-	-	-	YES	-	YES	YES
Instruction affects:	-	-	-	-	-	YES	YES	YES	1

O	1)	OR	1/2	0.10	0.06	0.05	0.03	0.004
ON	1)	OR NOT						
X	1)	EXCLUSIVE OR						
XN	1)	EXCLUSIVE OR						

Status word for: **O , ON , X , XN**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:	-	-	-	-	-	-	-	YES	YES
Instruction affects:	-	-	-	-	-	0	YES	YES	1

1) For valid operands and parameter ranges (see address types, page 14); Timers and counters (see page 25)

Bit Logic Instructions with Parenthetical Expressions

Saving the BR, RLO and OR bits and a function identifier (A, AN, ...) to the nesting stack. Seven nesting levels are possible per block. The listed parenthesese also apply to the “right parenthesis”-Instructions.

Instruction	Description	Length in Words	Typical Execution Time in µs				
			312	314, 151	315, 154	317	319
A(	AND left parenthesis	1	0.28	0.15	0.12	0.05	0.013
AN(	AND NOT left parenthesis						
O(	OR left parenthesis						
ON(	OR NOT left parenthesis						
X(	EXCLUSIVE OR left parenthesis						
XN(	EXCLUSIVE OR NOT left parenthesis						

Status word for: **A(, AN(, O(, ON(, X(, XN(**	BR	CC1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:	Yes	-	-	-	-	Yes	-	Yes	Yes
Instruction affects:	-	-	-	-	-	0	1	-	0

Instruction	Description	Length in Words	Typical Execution Time in µs				
			312	314, 151	315, 154	317	319
)	Right parenthesis, popping an entry off the nesting stack, gating the RLO with the current RLO in the processor	1	0.28	0.15	0.12	0.05	0.013

Status word for:)	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	-	-	-	-	-	-	-	Yes	-
Instruction affects:	Yes	-	-	-	-	Yes	1	Yes	1

O	ORing of AND operations according to the rule: AND before OR	1	0.08	0.05	0.04	0.02	0.008

Status word for: O	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	-	-	-	-	-	Yes	-	Yes	Yes
Instruction affects:	-	-	-	-	-	Yes	1	-	Yes

Logic Instructions with Timers and Counters

Examining the signal state of the addressed timer/counter and gating the result with the RLO according to the appropriate logic function.

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs				
				312	314, 151	315, 154	317	319
A	T f [1)]	AND Timer	1/2	0.60	0.30	0.23	0.13	0.020
	Z f [1)]	AND Counter		0.30	0.12	0.10	0.05	0.010
AN	T f [1)]	AND NOT Timer		0.60	0.30	0.23	0.13	0.020
	Z f [1)]	AND NOT Counter		0.30	0.12	0.10	0.05	0.010

Status word for: **A, AN**	CC 1	BR	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	-	-	-	-	-	Yes	-	Yes	Yes
Instruction affects:	-	-	-	-	-	Yes	Yes	Yes	1

[1)] For valid parameter ranges (see address types, page 14)

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs				
				312	314, 151	315, 154	317	319
O	T f [1]	OR timer	1/2	0.60	0.30	0.23	0.13	0.020
	Z f [1]	OR counter		0.30	0.12	0.10	0.05	0.010
ON	T f [1]	OR NOT timer		0.60	0.30	0.23	0.13	0.020
	Z f [1]	OR NOT counter		0.30	0.12	0.10	0.05	0.010
X	T f [1]	EXCLUSIVE OR timer		0.60	0.30	0.23	0.13	0.020
	Z f [1]	EXCLUSIVE OR counter		0.30	0.12	0.10	0.05	0.010
XN	T f [1]	EXCLUSIVE OR NOT timer		0.60	0.30	0.23	0.13	0.020
	Z f [1]	EXCLUSIVE OR NOT counter		0.30	0.12	0.10	0.05	0.010

Status word for: **O,ON,X,XN**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	-	-	-	-	-	-	-	Yes	Yes
Instruction affects:	-	-	-	-	-	0	Yes	Yes	1

[1] For valid parameter ranges (see address types, page 14)

Evaluating Conditions Using AND, OR and EXCLUSIVE OR

Examining the specified conditions for their signal status, and gating the result with the RLO according to the appropriate function.

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs				
				312	314, 151	315, 154	317	319
A		AND						
O		OR						
X		EXCLUSIVE OR						
	== 0	Result=0 (CC 1=0)and (CC 0=0)	1	0.30	0.09	0.08	0.03	0.010
	>0	Result>0 (CC 1=1) and (CC 0=0)						
	<0	Result<0 (CC 1=0)and (CC 0=1)						
	<>0	Result ≠ 0 ((CC1=0)and(CC 0=1)or (CC1=1)and(CC 0=0))						
	<=0	R<=0((CC 1=0) and (CC 0=1) or (CC1=0) and (CC 0=0))						
	>=0	R>=0((CC 1=1) and (CC 0=0) or (CC1=0) and (CC 0=0))						
	UO	AND unordered math instruction (CC 1=1) and (CC 0=1)						
	OS	AND OS=1						
	BR	AND BR=1						
	OV	AND OV=1						

Status word for: **A, O, X**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	Yes	Yes	Yes	Yes	Yes	Yes	-	Yes	Yes
Instruction affects:	-	-	-	-	-	Yes	Yes	Yes	1

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs				
				312	314, 151	315, 154	317	319
AN		AND NOT						
ON		OR NOT						
XN		EXCLUSIVE OR NOT						
	== 0	Result=0 (CC 1=0)and (CC 0=0)	1	0.30	0.09	0.08	0.03	0.010
	>0	Result>0 (CC 1=1) and (CC 0=0)						
	<0	Result<0 (CC 1=0)and (CC 0=1)						
	<>0	Result ≠ 0 ((CC1=0)and(CC 0=1)or (CC1=1)and(CC 0=0))						
	<=0	R<=0((CC 1=0) and (CC 0=1) or (CC1=0) and (CC 0=0))						
	>=0	R>=0((CC 1=1) and (CC 0=0) or (CC1=0) and (CC 0=0))						
	UO	AND unordered math instruction (CC 1=1) and (CC 0=1)						
	OS	AND OS=1						
	BR	AND BR=1						
	OV	AND OV=1						

Status word for: **AN, ON, XN**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	Yes	Yes	Yes	Yes	Yes	Yes	-	Yes	Yes
Instruction affects:	-	-	-	-	-	Yes	Yes	Yes	1

Edge-Triggered Instructions

Detection of an edge change. The current signal state of the RLO is compared with the signal state of the instruction or “edge bit memory”. FP detects a change in the RLO from “0” to “1”; FN detects a change in the RLO from “1” to “0”.

In-struc-tion	Address Iden-tifier	Description	Length in Words	Typical Execution Time in µs				
				312	314, 151	315, 154	317	319
FP	1)	Detecting the positive edge in the RLO.	2	0.26	0.17	0.15	0.08	0.015
FN	1)	Detecting the negative edge in the RLO.						

Status word for: **FP,FN**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	-	-	-	-	-	-	-	Yes	-
Instruction affects:	-	-	-	-	-	0	Yes	Yes	1

1) For all valid operands and parameter ranges (see address types, page 14)

Setting/Resetting Bit Addresses

Assigning the value “1” or “0” or the RLO o the addressed instruction. The instructions can be MCR-dependent.

In-struc-tion	Address Iden-tifier	Description	Length in Words	Typical Execution Time in µs				
				312	314, 151	315, 154	317	319
S	1)	Set input/output,bit memory,local data bit,data bit,instance data bit to “1”	2	0.14	0.09	0.08	0.04	0.010
R	1)	Set input/output,bit memory,local data bit,data bit,instance data bit to “0”						
=	1)	Assign RLO to input/output,bit memory,local data bit,data bit,instance data bit	2					

Status word for: **S, R; =**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	-	-	-	-	-	-	-	Yes	-
Instruction affects:	-	-	-	-	-	0	Yes	-	0

1) For all valid operands and parameter ranges (see address types, page 14)

Instructions Directly Affecting the RLO

The following instructions have a direct effect on the RLO.

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs				
				312	314, 151	315, 154	317	319
CLR		Set RLO to ”0”	2	0.07	0.05	0.04	0.02	0.004

Status word for: **CLR**	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	-	-	-	-	0	0	0	0

SET		Set RLO to ”1”	2	0.07	0.05	0.04	0.02	0.004

Status word for: **SET**	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	-	-	-	-	0	1	1	0

NOT		Negate RLO	2	0.07	0.05	0.04	0.02	0.004

Status word for: **NOT**	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
Instruction depends on:	-	-	-	-	-	Yes	-	Yes	-
Instruction affects:	-	-	-	-	-	-	1	Yes	-

SAVE		Retain the RLO in the Bit BR	2	0.08	0.05	0.04	0.02	0.004

Status word for: **SAVE**	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
Instruction depends on:	-	-	-	-	-	-	-	Yes	-
Instruction affects:	Yes	-	-	-	-	-	-	-	-

Timer Instructions

Starting or resetting a timer (addressed directly or via a parameter). The time value must be in ACCU1-L.

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs				
				312	314, 151	315, 154	317	319
SI	T f [1)]	Start timer as pulse on edge change from “0” to “1”	4/6	1.20	0.63	0.48	0.19	0.075
SV	T f [1)]	Start timer as exded pulse on edge change from “0” to “1”		1.11	0.57	0.46	0.18	0.065
SE	T f [1)]	Start timer as ON delay on edge change from “0” to “1”		1.31	0.69	0.53	0.21	0.080
SS	T f [1)]	Start timer as retive ON delay on edge change from “0” to “1”		1.25	0.66	0.51	0.20	0.070
SA	T f [1)]	Start timer as off-delay timer when the edge changes from “1” to “0”.		1.37	0.72	0.55	0.21	0.080
FR	T f [1)]	Enable timer for restarting on edge change from “0” to “1” (reset edge bit memory for starting timer		1.28	0.67	0.52	0.20	0.060
R	T f [1)]	Reset timer		1.51	0.79	0.61	0.24	0.115

Status word for: **SI, SV, SE, SS, SA, FR, R**	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
Instruction depends on:	-	-	-	-	-	-	-	Yes	-
Instruction affects:	-	-	-	-	-	0	-	-	0

[1)] For valid parameter ranges (see address types, page 16)

Counter Instructions

The count value is in ACCU1-L or in the address transferred as parameter.

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs				
				312	314, 151	315, 154	317	319
S	T f [1)]	Presetting of counter on edge change from “0” to “1”	4/6	1.76	0.92	0.71	0.28	0.090
R	T f [1)]	Reset counter to “0” on edge change from “0” to “1”		1.15	0.60	0.46	0.17	0.050
CU	T f [1)]	Increment counter by 1 on edge change from “0” to “1”		1.22	0.64	0.49	0.20	0.055
CD	T f [1)]	Decrement counter by 1 on edge change from “0” to “1”		1.31	0.69	0.53	0.20	0.060
FR	T f [1)]	Enable counter on edge change from “0” to “1” (reset edge bit memory for up and down counting)	2	1.19	0.62	0.48	0.19	0.055

Status word for: **S,R,CU,CD,FR**	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
Instruction depends on:	-	-	-	-	-	-	-	Yes	-
Instruction affects:	-	-	-	-	-	0	-	-	0

[1)] For all valid operands and parameter ranges (see address types, page 14)

Load Instructions

Loading address identifiers into ACCU1. The conts of ACCU1 and ACCU2 are saved first. The status word is not affected.

In-struc-tion	Address Identi-fier	Description	Length in Words	Typical Execution Time in µs				
				312	314, 151	315, 154	317	319
L		Load ...						
	B [1]	Byte	1/2	0.24	0.12	0.09	0.03	0.070
	W [1]	Word		0.28	0.14	0.11	0.04	0.010
	DW [1]	Double word		0.32	0.16	0.12	0.04	0.015
	k8 [2]	8-bit constant in ACCU1-LL	1	0.24	0.12	0.09	0.03	0.070
	k16 [2]	16-bit constant in ACCU1-L	2					
	k32 [2]	32-bit constant in ACCU1	3					

[1] For all valid operands and parameter ranges (see address types, page 14)
[2] Valid for all constants (see page 8)

Load Instructions for Timers and Counters

Loading a time value or count value into ACCU1. The contents of ACCU1 are first saved to ACCU2. The bits of the status word are not affected.

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs				
				312	314, 151	315, 154	317	319
L	T f [1)]	Load time value	1/2	1.70	0.80	0.80	0.34	0.125
LC	T f [1)]	Load time value in BCD		2.71	1.41	1.09	0.43	0.280
L	Z f [1)]	Load count value		1.11	0.58	0.45	0.14	0.050
LC	Z f [1)]	Load count value in BCD		1.71	0.89	0.69	0.27	0.155

[1)] For valid parameter ranges (see address types, page 14)

Transfer Instructions

Transferring the contents of ACCU1 to the addressed Inrand. The status word is not affected. Remember that some transfer instructions depend on the MCR.

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs				
				312	314, 151	315, 154	317	319
T		Transfer contents of...						
	B [1)]	ACCU1-LL to ...input byte	1/2	0.20	0.10	0.08	0.03	0.007
	W [1)]	ACCU1-L to ... input word		0.24	0.12	0.09	0.03	0.008
	DW [1)]	ACCU1 to ...input double word		0.28	0.14	0.11	0.04	0.010

[1)] For all valid operands and parameter ranges (see address types, page 14)

Load and Transfer Instructions for Address Registers

Loading a double word from a memory area or register into AR1 or AR2.

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs				
				312	314, 151	315, 154	317	319
LAR1		Load contents from ... into AR1						
	-	ACCU1...	1	0.20	0.10	0.10	0.03	0.010
	AR2	Address register 2...	1	0.20	0.10	0.10	0.03	0.010
	DBD a	Data double word...	2	0.51	0.27	0.21	0.08	0.020
	DID a	Instance data double word...	2	0.98	0.51	0.40	0.15	0.050
	m	32-bit constant as pointer...	3	0.30	0.15	0.12	0.04	0.010
	LD a	Local data double word...	2	0.51	0.27	0.21	0.08	0.020
	MD a	Bit memory double word...	2	0.51	0.27	0.21	0.08	0.020
LAR2		Load contents from ... into AR2						
	-	ACCU1...	1	0.20	0.10	0.10	0.03	0.010
	DBD a	Data double word...	2	0.51	0.27	0.21	0.08	0.020
	DID a	Instance data double word...	2	0.98	0.51	0.40	0.15	0.050
	m	32-bit constant as pointer...	3	0.30	0.15	0.12	0.04	0.010
	LD a	Local data double word...	2	0.51	0.27	0.21	0.08	0.020
	MD a	Bit memory double word...	2	0.51	0.27	0.21	0.08	0.020

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs				
				312	314, 151	315, 154	317	319
TAR1		Transfer contents of AR1 to ...						
	-	ACCU1	1	0.30	0.16	0.13	0.04	0.020
	AR2	Address register 2	1	0.20	0.10	0.10	0.03	0.010
	DBD a	Data double word	2	0.39	0.21	0.17	0.06	0.020
	DID a	Instance data double word	2	0.93	0.49	0.38	0.14	0.045
	LD a	Local data double word	2	0.39	0.21	0.17	0.06	0.020
	MD a	Bit memory double word...	2	0.39	0.21	0.17	0.06	0.020
TAR2		Transfer contents of AR2 to ...						
	-	ACCU1	1	0.30	0.16	0.13	0.04	0.020
	DBD a	Data double word	2	0.39	0.21	0.17	0.06	0.020
	DID a	Instance data double word	2	0.93	0.49	0.38	0.14	0.045
	LD a	Local data double word	2	0.39	0.21	0.17	0.06	0.020
	MD a	Bit memory double word	2	0.39	0.21	0.17	0.06	0.020
TAR		Exchange the contents of AR1 and AR2	1	0.28	0.16	0.13	0.04	0.010

Load and Transfer Instructions for the Status Word

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs				
				312	314, 151	315, 154	317	319
L	STW	Load status word [1] into ACCU1	1	0.63	0.33	0.26	0.09	0.025

Status word for: **L STW**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	Yes	Yes	Yes	Yes	Yes	0	0	Yes	0
Instruction affects:	-	-	-	-	-	-	-	-	-

T	STW	Transfer ACCU1(bits 0 to 8) to the status word [1]	1	0.58	0.31	0.24	0.09	0.020

Status word for: **T STW**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	Yes	Yes	Yes	Yes	Yes	-	-	Yes	-

[1] Structure of the status word, see page 13

Load Instructions for DB Number and DB Length

Loading the number/length of a data block into ACCU1. The old contents of ACCU1 are saved to ACCU2. The condition code bits are not affected.

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs				
				312	314, 151	315, 154	317	319
L	DBNO	Load number of data block	1	0.27	0.15	0.12	0.04	0.010
L	DINO	Load number of instance data block						
L	DBLG	Load length of data block into byte	1	0.34	0.19	0.14	0.04	0.010
L	DILG	Load length of instance data block into byte						

Word Logic Instructions with the Contents of Accumulator 1

Gating the contents of ACCU1 and/or ACCU1-L with a word or double word according to the appropriate function. The word or double word is either a constant in the instruction or in ACCU2. The result is in ACCU1 and/or ACCU1-L.

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs				
				312	314, 151	315, 154	317	319
AW		AND ACCU2-L	1	0.33	0.18	0.14	0.05	0.014
OW		OR ACCU2-L						
XOW		EXCLUSIVE OR ACCU2-L						
AW	k16	AND 16-bit constant	2	0.33	0.18	0.14	0.05	0.014
OW	k16	OR 16-bit constant						
XOW	k16	EXCLUSIVE OR 16-bit constant						

Status word for: **AW, OW, XOW**	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	Yes	0	0	-	-	-	-	-

Instruction	Address Identifier	Description	Length in Word	Typical Execution Time in µs				
				312	314, 151	315, 154	317	319
AD		AND ACCU2	1	0.28	0.16	0.13	0.05	0.014
OD		OR ACCU2						
XOD		EXCLUSIVE OR ACCU2						
AD	k32	AND 32-bit constant	3	0.28	0.16	0.13	0.05	0.014
OD	k32	OR 32-bit constant						
XOD	k32	EXCLUSIVE OR 32-bit constant						

Status word for: **AD, OD, XOD**	Br	CC1	CC0	OV	OS	OR	STA	RLO	FC
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	Yes	0	0	-	-	-	-	-

Fixed-point arithmetic (16/32 bit) / Floating-point arithmetic (32 bit)

Mathematical functions of two 16/32-bit numbers. The result is in ACCU1 or ACCU1-L.

I = Integer -> 16-bit, D = Integer -> 32-bit, R = Real number -> 32-bit

In-struc-tion	Description	Length in Words	Typical Execution Time in µs				
			312	314, 151	315, 154	317	319
	Add 2 integers or real numbers	1					
+I	+I: (ACCU1-L)=(ACCU1-L)+(ACCU2-L)		0.25	0.13	0.10	0.04	0.010
+D	+D (ACCU1)=(ACCU2)+(ACCU1)		0.22	0.12	0.09	0.03	0.010
+R	+R (ACCU1)=(ACCU2)+(ACCU1)		1.10	0.58	0.44	0.16	0.040
	Subtract 2 integers or real numbers						
-I	-I: (ACCU1-L)=(ACCU2-L)+(ACCU1-L)		0.25	0.13	0.14	0.04	0.010
-D	-D: (ACCU1)=(ACCU2) - (ACCU1)		0.22	0.12	0.09	0.03	0.010
-R	-R: (ACCU1)=(ACCU2) - (ACCU1)		1.10	0.58	0.44	0.16	0.040

Status word for: **+ I,+ D, +R; - I, - D, - R**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{\text{FC}}$
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	Yes	Yes	Yes	Yes	-	-	-	-

In-struc-tion	Description	Length in Words	Typical Execution Time in µs				
			312	314, 151	315, 154	317	319
	Multiply 2 integers or real numbers	1					
*I	*I: (ACCU1)=(ACCU2-L) * (ACCU1-L)		0.28	0.15	0.12	0.04	0.010
*D	*D: (ACCU1)=(ACCU2) *(ACCU1)		0.21	0.12	0.09	0.03	0.008
*R	*R: (ACCU1)=(ACCU2) *(ACCU1)		1.11	0.58	0.44	0.16	0.040
	Divide 2 integers or real numbers						
/I	/I: (ACCU1-L)=(ACCU2-L):(ACCU1-L) --> The remainder of the division is in the		0.52	0.27	0.22	0.08	0.060
/D	ACCU1-H		0.51	0.27	0.21	0.08	0.050
/R	/D: (ACCU1)=(ACCU2): (ACCU1) /R: (ACCU1)=(ACCU2): (ACCU1)		4,85	2,52	1.93	0.25	0.060
MOD	Divide 2 integers (32-bit) and load the remainder of the division in ACCU1: (ACCU1)= Remainder of [(ACCU2):(ACCU1)]		0.43	0.23	0.18	0.07	0.060

Status word for: ***I, * D, * R;** / **I,** / **D,** / **R; MOD**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{\text{FC}}$
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	Yes	Yes	Yes	Yes	-	-	-	-

In-struc-tion	Description	Length in Words	Typical Execution Time in µs				
			312	314, 151	315, 154	317	319
NEGR	Negate the real number in ACCU1	1	0.20	0.12	0.09	0.03	0.005
ABS	Form the absolute value of the real number in ACCU1		0.20	0.12	0.09	0.03	0.005

Status word for: **NEGR, ABS**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	-	-	-	-	-	-	-	-

Square root, Square (32-bit) / Logarithm function (32-bit)

The result of the instruction / logarithm function is in ACCU1. The instructions can be interrupted by interrupts.

Instruction	Description	Length in Words	Typical Execution Time in µs				
			312	314, 151	315, 154	317	319
SQRT	Calculate the square root of a real number in ACCU1	1	8.14	4.22	3.24	1.26	0.475
SQR	Form the square of a real number in ACCU1		1.15	0.59	0.46	0.18	0.040
LN	Form the natural logarithm of a real number in ACCU1	1	7.34	3.80	2.92	1.20	0.455
EXP	Calculate the exponential value of a real number in ACCU1 to the base e (= 2.71828)		9.13	4.73	3.63	1.50	0.525

Status word for: **SQRT, SQR, LN, EXP**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	Yes	Yes	Yes	Yes	-	-	-	-

Trigonometrical Functions (32 Bits)

The result of the instruction is in ACCU1. The instructions can be interrupted.

Instruction	Description	Length in Word	Typical Execution Time in µs				
			312	314, 151	315, 154	317	319
SIN 1)	Calculate the sine of a real number	1	7.25	3.90	3.00	1.20	0.530
ASIN 2)	Calculate the arcsine of a real number	1	15.84	8.40	6.44	1.30	0.480
COS 1)	Calculate the cosine of a real number	1	9.19	4.75	3.65	1.50	0.530
ACOS 2)	Calculate the arccosine of a real number	1	7.21	3.73	2.87	1.20	0.450
TAN 1)	Calculate the tangent of a real number	1	10.92	5.97	4.35	1.80	0.620
ATAN 2)	Calculate the arctangent of a real number	1	7.91	4.10	3.14	1.30	0.485

Status word for: **SIN, ASIN, COS, ACOS, TAN, ATAN**	BR	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{\text{FC}}$
Instruction depends on:		-	-	-	-	-	-	-	-	-
Instruction affects:		-	Yes	Yes	Yes	Yes	-	-	-	-

1) Specify the angle in radians; the angle must be given as a floating point value in ACCU 1.
2) The result is an angle in radians.

Adding Constants

Adding integer constants and storing the result in ACCU1. The condition code bits are not affected.

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs				
				312	314, 151	315, 154	317	319
+	i8	Add an 8-bit integer constant	1	0.20	0.10	0.10	0.05	0.010
+	i16	Add a 16-bit integer constant	2	0.20	0.10	0.10	0.05	0.010
+	i32	Add a 32-bit integer constant	3	0.20	0.10	0.10	0.05	0.010

Adding Using Address Registers

Adding a 16-bit integer to the contents of the address register. The value is in the instruction or in ACCU1-L. The condition code bits are not affected.

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs				
				312	314, 151	315, 154	317	319
+AR1	-	Add the contents of ACCU1-L to those of AR1	1	0.20	0.10	0.10	0.07	0.010
+AR1	m	Add a pointer constant to the contents of AR1	2	0.40	0.15	0.12	0.07	0.010
+AR2	-	Add the contents of ACCU1-L to those of AR2	1	0.20	0.10	0.10	0.07	0.010
+AR2	m	Add pointer constant to the contents of AR2	2	0.40	0.15	0.12	0.07	0.010

Comparison Instructions with Integers (16-bit / 32-bit) or with 32-bit real numbers

Comparison of integers (16-bit) in ACCU1-L and ACCU2-L. RLO=1. if the condition is satisfied.
Comparison of integers (32-bit) in ACCU1 and ACCU2. RLO=1. if the condition is satisfied.
Comparison of 32-bit real numbers in ACCU1 and ACCU2. RLO=1. if the condition is satisfied.

In-struc-tion	Description	Length in Words	Typical Execution Time in µs				
			312	314, 151	315, 154	317	319
==I	==I: ACCU2-L=ACCU1-L	1	0.48	0.26	0.20	0.07	0.028
==D	==D: ACCU2=ACCU1		0.43	0.23	0.18	0.06	0.023
==R	==R: ACCU2=ACCU1		1.67	0.87	0.67	0.27	0.046
<>I	<>I: ACCU2-L≠ACCU1-L		0.48	0.26	0.20	0.07	0.028
<>D	<>D: ACCU2≠ACCU1		0.43	0.23	0.18	0.06	0.023
<>R	<>R: ACCU2≠ACCU1		1.67	0.87	0.67	0.27	0.046
<I	<I: ACCU2-L<ACCU1-L		0.48	0.26	0.20	0.07	0.028
<D	<D: ACCU2<ACCU1		0.43	0.23	0.18	0.06	0.023
<R	<R: ACCU2<ACCU1		1.67	0.87	0.67	0.27	0.046
<=I	<=I: ACCU2-L<=ACCU1-L		0.48	0.26	0.20	0.07	0.028
<=D	<=D: ACCU1<=ACCU2		0.43	0.23	0.18	0.06	0.023
<=R	<=R: ACCU1<=ACCU2		1.67	0.87	0.67	0.27	0.046

Instruction	Description	Length in Words	Typical Execution Time in µs				
			312	314, 151	315, 154	317	319
>I >D >R	>I: ACCU2-L>ACCU1-L >D: ACCU2>ACCU1 >R: ACCU2>ACCU1	1	0.48 0.43 1.67	0.26 0.23 0.87	0.20 0.18 0.67	0.07 0.06 0.27	0.028 0.023 0.046
>=I >=D >=R	>I: ACCU2-L>=ACCU1-L >D: ACCU2>=ACCU1 >R: ACCU2>=ACCU1		0.48 0.43 1.67	0.26 0.23 0.87	0.20 0.18 0.67	0.07 0.06 0.27	0.028 0.023 0.046

Status word for: **== I, D; <> I, D; < I, D; <= I, D; > I, D; >= I, D**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	Yes	Yes	0	-	0	Yes	Yes	1

Status word for: **==R, <>R, <R, <=R, >R, >=R**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	Yes	Yes	Yes	Yes	0	Yes	Yes	1

Shift Instructions

Shifting the contents of ACCU1 and ACCU1-L to the left or right by the specified number of places. If no address identifier is specified, shift the number of places into ACCU2-LL. Any positions that become free are padded with zeros or the sign. The last bit shifted is in condition code bit CC 1.

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs				
				312	314, 151	315, 154	317	319
SLW	- 0 ... 15	Shift the contents of ACCU1-L to the left. Positions that become free are provided with zeros.	1	0.51	0.27	0.21	0.08	0.019
SLD	- 0 ... 32	Shift the contents of ACCU1 to the left. Positions that become free are provided with zeros.		0.46	0.24	0.19	0.07	0.019
SRW	- 0 ... 15	Shift the contents of ACCU1-L to the right. Positions that become free are provided with zeros.		0.51	0.27	0.21	0.08	0.019
SRD	- 0 ... 32	Shift the contents of ACCU1 to the right. Positions that become free are provided with zeros.		0.46	0.24	0.19	0.07	0.019
SSI	- 0 ... 15	Shift the contents of ACCU1-L with sign to the right. Positions that become free are provided with the sign (bit 15).		0.60	0.30	0.23	0.09	0.019
SSD	- 0 ... 32	Shift the contents of ACCU1 with sign to the right		0.46	0.27	0.19	0.08	0.019

Status word for: **SLW, SLD, SRW, SRD, SSI, SSD**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	Yes	Yes	Yes	-	-	-	-	-

Rotate Instructions

Rotate the contents of ACCU1 to the left or right by the specified number of places. If no address identifier is specified, rotate the number of places into ACCU2-LL.

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs				
				312	314, 151	315, 154	317	319
RLD	- 0 ... 32	Rotate the contents of ACCU1 to the left	1	0.45	0.24	0.19	0.07	0.019
RRD	- 0 ... 32	Rotate the contents of ACCU1 to the right		0.45	0.24	0.19	0.07	0.019

Status word for: **RLD, RRD**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	Yes	Yes	Yes	-	-	-	-	-

RLDA	-	Rotate the contents of ACCU1 one bit position to the left via condition code bit A1	1	0.30	0.16	0.13	0.05	0.012
RRDA	-	Rotate the contents of ACCU1 one bit position to the right via condition code bit A1		0.30	0.16	0.13	0.05	0.015

Status word for: **RLDA, RRDA**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	Yes	0	0	-	-	-	-	-

Accumulator Transfer Instructions, Incrementing and Decrementing

The status word is not affected.

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs				
				312	314, 151	315, 154	317	319
CAW	-	Reverse the order of the bytes in ACCU1-L. LL, LH becomes LH, LL.	1	0.2	0.10	0.10	0.05	0.010
CAD	-	Reverse the order of the bytes in ACCU1. LL, LH, HL, AA becomes HH, HL, LH, LL.	1	0.4	0.20	0.16	0.06	0.010
TAK	-	Swap the contents of ACCU1 and ACCU2	1	0.25	0.14	0.11	0.04	0.010
PUSH	-	The contents of ACCU1 are transferred to ACCU2.	1	0.20	0.10	0.08	0.03	0.010
POP	-	The contents of ACCU2 are transferred to ACCU1:	1	0.20	0.10	0.08	0.03	0.010
INC	0 ... 255	Increment ACCU1-LL	1	0.20	0.10	0.10	0.05	0.010
DEC	0 ... 255	Decrement ACCU1-LL	1	0.20	0.10	0.10	0.05	0.010

Program Display and Null Operation Instructions

The status word is not affected.

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs				
				312	314, 151	315, 154	317	319
BLD[1)]	0 ... 255	Program display instruction: Is treated by the CPU like a null operation instruction.	1	0.00	0.00	0.00	0.00	0.00
NOP[1)]	0 1	Null Operation instruction		0.00	0.00	0.00	0.00	0.00

1) The BLD instructions are generated and used by the programming device and should not be deleted, changed or added to. The NOP1 instructions should not be used. If you require a NOP instruction, use NOP0.

Data Type Conversion Instructions

The results of the conversion are in ACCU1. When converting real numbers, the execution time depends on the value.

Instruction	Description	Length in Words	Typical Execution Time in µs				
			312	314, 151	315, 154	317	319
BTI	Conv. cont. of ACCU1 from BCD to integer (16 bits) (**B**CD **T**o **I**nt)	1	0.73	0.39	0.30	0.11	0.040
BTD	Conv. cont. of ACCU1 from BCD to double int. (32 bits) (**B**CD **T**o **D**oubleint)	1	1.08	0.57	0.44	0.16	0.090
DTR	Convert contents of ACCU1 from double integer to real (32 bits) (**D**oubleint **T**o **R**eal)	1	0.70	0.37	0.29	0.11	0.020
ITD	Convert contents of ACCU1 from integer (16 bits) to double int. (32 bits) (**I**nt **T**o **D**oubleint)	1	0.21	0.10	0.09	0.03	0.008

Status word for: **BTI, BTD, DTR, ITD**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{\text{FC}}$
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	-	-	-	-	-	-	-	-

Instruction	Description	Length in Words	Typical Execution Time in µs				
			312	314, 151	315, 154	317	319
ITB	Conv. cont. of ACCU1 from int. (16 bits) to BCD from 0 to +/- 999 (**I**nt **T**o **B**CD)	1	1.09	0.57	0.44	0.17	0.117
DTB	Conv. cont. of ACCU1 f. double int. (32 bits) t. BCD f. 0 to +/-9 999 999 (**D**oubleint **T**o **B**CD)		2.98	1.54	1.19	0.47	0.315
RND	Convert a real number into a 32-bit integer.		4.82	2.49	1.92	0.15	0.025
RND-	Convert a real number into a 32-bit integer. The number is rounded to the next whole number.		4.82	2.49	1.92	0.15	0.025
RND+	Convert a real number into a 32-bit integer. The number is rounded to the next whole number.		4.82	2.49	1.92	0.15	0.025
TRUNC	Convert a real number into a 32-bit integer. The places after the decimal point are truncated.		4.82	2.49	1.92	0.15	0.025

Status word for: **ITB, DTB, RND, RND- , RND+,TRUNC**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	-	-	Yes	Yes	-	-	-	-

Forming the Ones and Twos Complements

Instruction	Description	Length in Words	Typical Execution Time in µs				
			312	314, 151	315, 154	317	319
INVI	Form the ones complement of ACCU1-L	1	0.13	0.08	0.07	0.04	0.010
INVD	Form the ones complement of ACCU1		0.11	0.07	0.06	0.03	0.005

Status word for: **INVI, INVD**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	-	-	-	-	-	-	-	-

NEGI	Form the twos complement of ACCU1-L (integer)	1	0.16	0.10	0.08	0.05	0.010
NEGD	Form the twos complement of ACCU1 (double integer)		0.12	0.07	0.06	0.03	0.005

Status word for: **NEGI, NEGD**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	Yes	Yes	Yes	Yes	-	-	-	-

Block Call Instructions

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs				
				312	314, 151	315, 154	317	319
CALL	FB q, DB q	Unconditional call of an FB, with parameter transfer	1	5.10	2.65	2.05	0.78	0.350
CALL	SFB q, DB q	Unconditional call of an SFB, with parameter transfer	2	1)	1)	1)	1)	1)
CALL	FC q	Unconditional call of a function, with parameter transfer	1	4.87	2,59	2.03	0.83	0.350
CALL	SFC q	Unconditional call of an SFC, with parameter transfer	2	1)	1)	1)	1)	1)

Status word for: **CALL**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	-	-	-	0	0	1	-	0

1) In the chapter: System functions, page 92 or system function blocks, page 104

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs				
				312	314, 151	315, 154	317	319
UC	FB q	Unconditional call of blocks without parameter transfer	1	3.97	2.06	1.59	0.62	0.300
	FC q			4.26	2.27	1.77	0.72	0.300
	Parameter	FB/FC call via parameter		4.26	2.27	1.77	0.72	0.300
CC	FB q	Conditional call of blocks without parameter transfer	1	3.97	2.06	1.59	0.62	0.300
	FC q			4.26	2.27	1.77	0.72	0.300
	Parameter	FB/FC call via parameter		4.26	2.27	1.77	0.72	0.300

Status word for: **UC, CC**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	-	-	-	0	0	1	-	0

OPN [2]	DB q	Open data block	1/2 [1]	0.40	0.21	0.17	0.08	0.020
	DI q	Open instance data block	2	0.40	0.21	0.17	0.08	0.020
	Parameter	Open instance data block	2	0.40	0.21	0.17	0.08	0.020

Status word for: **OPN**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	-	-	-	-	-	-	-	-

1) For long block numbers (> 255)
2) The CPUs support symbolic programming. The fully qualified DB accesses (e.g. DB100.DBX 1.2) supported here generally cause no additional runtimes. This applies also for the ON DB command contained in the access.

Block End Instructions

Instruction	Description	Length in Words	Typical Execution Time in µs				
			312	314, 151	315, 154	317	319
BE	End block	1	1.2	0.88	0.68	0.26	0.070
BEU	End block unconditionally	1	1.2	0.88	0.68	0.26	0.070

Status word for: **BE, BEU**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	-	-	-	0	0	1	-	0

BEC	End block conditionally if RLO = “1”	1	1.2	0.88	0.68	0.26	0.070

Status word for: **BEC**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	-	-	-	-	-	-	-	Yes	-
Instruction affects:	-	-	-	-	Yes	0	1	1	0

Exchanging Shared Data Block and Instance Data Block

Exchanging the two current data blocks. The current shared data block becomes the current instance data block, and vice versa. The condition code bits are not affected.

Instruction	Description	Length in Words	Typical Execution Time in µs				
			312	314, 151	315, 154	317	319
CDB	Exchange shared data block and instance data block	1	0.20	0.10	0.10	0.10	0.050

Jump Instructions

Jumping as a function of conditions. With 8-bit operands the jump width is between -128 and +127. In the case of 16-bit operands, the jump width lies between -32768 and -129 (+128 and +32767).

Note:
Please note for S7-300 CPU programs that the jump operations starting from a logic string or into a logic string are invalid.

Operations that set the /ER=0 indicate the end of a logic string.

The beginning is the first logic operation after the end of a logic string. Here the linear program sequence is relevant without taking into consideration the jump operations.

Please note that the operation AND before OR also indicates the beginning of a new logic string. In the same manner, jump operations into a different nesting level are invalid.

You can find examples starting on page 66.

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs				
				312	314, 151	315, 154	317	319
JC	LABEL	Jump if RLO ="1"	1 [1]/2	0.39	0.21	0.16	0.10	0.010
JCN	LABEL	Jump if RLO ="0"	2	0.39	0.21	0.16	0.10	0.010

Status word for: **JC, JCN**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	-	-	-	-	0	1	1	0

JCB	LABEL	Jump if RLO ="1"; Save the RLO in the BR bit	2	0.39	0.21	0.16	0.10	0.010
JNB	LABEL	Jump if RLO ="0"; Save the RLO in the BR bit	2	0.39	0.21	0.16	0.10	0.010

Status word for: **JCB, JNB**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	yes	-	-	-	-	0	1	1	0

JBI	LABEL	Jump if BR ="1"	2	0.39	0.21	0.16	0.10	0.010
JBIN	LABEL	Jump if BR ="0"	2	0.39	0.21	0.16	0.10	0.010

Status word for: **JBI, JBIN**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	yes	-	-	-	-	-	-	-	-
Instruction affects:	-	-	-	-	-	0	1	-	0

JO	LABEL	Jump on stored overflow (OV="1")	1 [1]/2	0.39	0.21	0.16	0.10	0.010

Status word for: **JO**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	-	-	-	yes	-	-	-	-	-
Instruction affects:	-	-	-	-	-	-	-	-	-

[1] 1 word long for jump length of -128 to +128

[1] 1 word long for jump length of -128 to +128

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs				
				312	314, 151	315, 154	317	319
JOS	LABEL	Jump on stored overflow (OS="1")	2	0.39	0.21	0.16	0.10	0.010

Status word for: **JOS**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	-	-	-	-	yes	-	-	-	-
Instruction affects:	-	-	-	-	-	-	-	-	-

JUO	LABEL	Jump if "unordered instruction" (CC 1=1 and CC 0=1)	2	0.39	0.21	0.16	0.10	0.010
JZ	LABEL	Jump if result =0 (CC 1=0 and CC 0=0)	1 [1]/2	0.39	0.21	0.16	0.10	0.010
JP	LABEL	Jump if result >0 (CC 1=1 and CC 0=0)		1 [1]/2	0.39	0.21	0.16	0.10
JM	LABEL	Jump if result <0 (CC 1=0 and CC 0=1)		1 [1]/2	0.39	0.21	0.16	0.10

Status word for: **JUO, JZ, JP, JM**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	-	yes	yes	-	-	-	-	-	-
Instruction affects:	-	-	-	-	-	-	-	-	-

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs				
				312	314, 151	315, 154	317	319
JN	LABEL	Jump if result00; (CC 1=1 and CC 0=0) or (CC 1=0) and (CC 0=1)	1 [1]/2	0.39	0.21	0.16	0.10	0.010
JMZ	LABEL	Jump if resultv0; (CC 1=0 and CC 0=1) or (CC 1=0) and (CC 0=0)	2	0.39	0.21	0.16	0.10	0.010
JPZ	LABEL	Jump if resultw0; (CC 1=1 and CC 0=0) or (CC 1=0) or (CC 1=0) and (CC 0=0)	2	0.39	0.21	0.16	0.10	0.010

Status word for: **JN, JMZ, JPZ**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	-	yes	yes	-	-	-	-	-	-
Instruction affects:	-	-	-	-	-	-	-	-	-

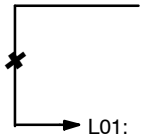
JU	LABEL	Jump unconditionally	1 [1]/2	0.39	0.21	0.16	0.10	0.010
JL	LABEL	Jump distributor The instruction is followed by a list of jump instructions. The operand is a jump lab to subsequent instructions in this list. ACCU-L contains the number of the jump instruction to be executed.	2	0.39	0.21	0.16	0.10	0.032
LOOP	LABEL	Decrement ACCU1-L and jump if ACCU1-L00 (loop programming)	2	0.35	0.19	0.15	0.06	0.010

Status word for: **JU, JL, LOOP**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	-	-	-	-	-	-	-	-

[1] 1 word long for jump length of -128 to +128

Examples of jump operations

// **Example 1: Invalid jump over the end of a logic string**

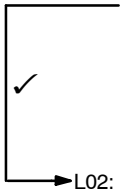


```
      =   M   10.0      // End of a logic string 1  1

      U   M    0.0      // Beginning of the logic string 2
      SPO L01           // Jump is invalid because it over jumps the end of the logic string
      U   M    0.1
      =   M   10.1      // End of the logic string 2

L01:  U   M    2.0      // Beginning of the logic string 3
      =   M   20.0      // End of the logic string  3
```

// **Example 2: Invalid jump at the end of a logic string**



```
      =   M   10.0      // End of a logic string 1

      U   M    0.0      // Beginning of the logic string 2
      SPB L02           // End of the logic string 2 since the SPB set the status bit /ER=0.
                           Jump in valid because it is located at the end of the logic string.

      U   M    0.1      // Beginning of the logic string 3
      =   M   10.1      // End of the logic string 3

L02:  U   M    2.0      // End of the logic string 4
      =   M   20.0      // End of the logic string 4
```

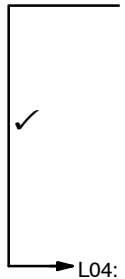
// **Example 3: Valid jump within a logic string**

```
      =   M   10.0      // End of the logic string 1

      U   M    0.0      // Beginning of the logic string 2
      SPO L03           // Jump within the logic string is valid. SPO does not close the logic string.
      U   M    0.1      // Logic operation
      U   M    0.2
L03:  U   M    0.3      // Jump on label within the logic string is valid
      U   M    0.4
      =   M   10.1      // End of the logic string 2

      U   M    2.0      // Beginning of the logic string 3
      =   M   20.0      // End of the logic string 3
```

// **Example 4: Valid jump over and past a logic string**



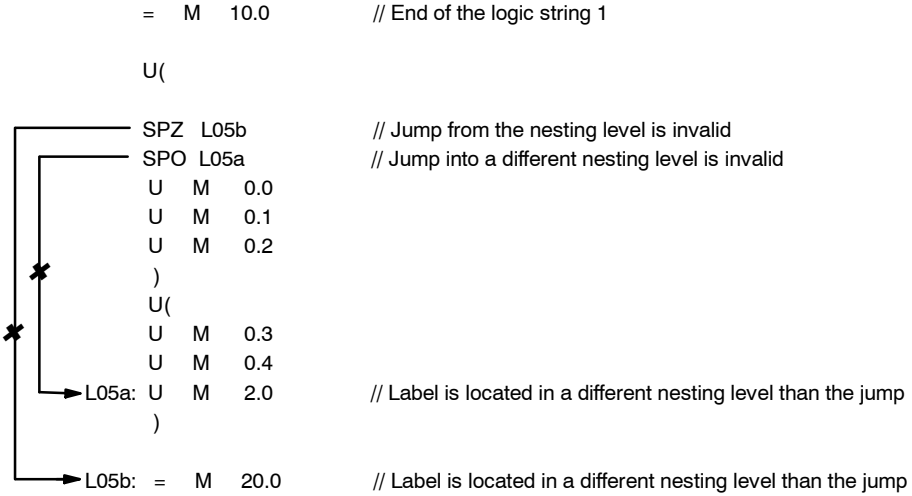
```
        =   M   10.0        // End of the logic string 1

        SPO L04             // Jump over and past the logic string is valid

        U   M   0.0         // Beginning of the logic string 2
        U   M   0.1         // Logic operation
✓       U   M   0.2
        U   M   0.3
        U   M   0.4
        =   M   10.1        // End of the logic string 2

L04:    U   M   2.0         // Beginning of the logic string 3. Jump on label is valid
                                because the jump is not located within a logic string
        =   M   20.0        // End of the logic string 3
```

// **Example 5: Invalid jumps between nesting levels**



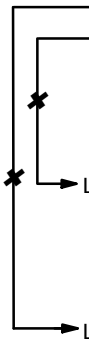
```
        =    M    10.0          // End of the logic string 1

        U(

        SPZ  L05b               // Jump from the nesting level is invalid
        SPO  L05a               // Jump into a different nesting level is invalid
        U    M    0.0
        U    M    0.1
        U    M    0.2
        )
        U(
        U    M    0.3
        U    M    0.4
L05a:   U    M    2.0          // Label is located in a different nesting level than the jump
        )

L05b:   =    M    20.0         // Label is located in a different nesting level than the jump
```

// **Example 6: Invalid jumps in AND before OR gatings**



```
      =   M   10.0        // End of the logic string 1

      U   M    0.0        // Beginning of the first AND before OR logic string
      SPZ L06b            // Jump from AND before OR logic string is invalid
      SPO L06a            // Jump into the second AND before OR logic string is invalid
      U   M    0.1
      U   M    0.2
      O                   // AND before OR operation
      U   M    0.3        // Beginning of the second AND before OR logic string
L06a: U   M    0.4        // Label is located in a different logic string than the jump
      U   M    0.5
      U   M    0.6
      =   M   20.0

L06b: U   M   20.0        // Label is located outside of the gating
```

Instructions for the Master Control Relay (MCR)

MCR=1→MCR is deactivated
MCR=0→MCR is activated; “T” and “=” instructions write “0” to the corresponding address identifiers; “S” and “R” instructions leave the memory contents unchanged.

Instruction	Description	Length in Words	Typical Execution Time in µs				
			312	315, 151	315, 154	317	319
MCR(	Open an MCR zone. Save the RLO to the MCR stack.	1	0.21	0.15	0.13	0.08	0.030
)MCR	Close an MCR-Zone. Pop an entry off the MCR-Stack.	1	0.21	0.15	0.13	0.08	0.030

Status word for: **MCR(,)MCR**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	-	-	-	-	-	-	-	Yes	-
Instruction affects:	-	-	-	-	-	0	1	-	0

MCRA	Activate the MCR	1	0.2	0.1	0.1	0.07	0.030
MCRD	Deactivate the MCR	1	0.2	0.1	0.1	0.07	0.030

Status word for: **MCRA, MCRD**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	-	-	-	-	-	-	-	-

Execution Times

You have to calculate the basic execution times for direct/indirect addressing. In this chapter we will explain this calculation to you.

Two-Part Statement

A statement consists of two parts:

Part 1: Load the address of the instruction (from page 22)

Part 2: Loading the address of the operand (from page 74)

This means that you also have to calculate the basic execution time of an instruction with addressed operand from these two parts.

Calculating the Execution Time

The total execution time is calculated as follows:

	Execution time of the instruction
+	Time required for loading the address
=	Total execution time of the instruction

The execution times listed in the chapter entitled “List of Instructions” apply to the execution times of the second part of an instruction, i.e. for the actual execution of an instruction.

You must then add the time required for loading the address of the instruction to this execution time (see Table on following page).

Loading the Addresses and Operands

Operand Range	Example	Typical Execution Time in µs				
		312	314, 151	315, 154	317	319
Direct addressing	L 1.234567e-36	0	0	0	0	0
I/O	U E a.b	0	0	0	0	0
M	U M a.b	0	0	0	0	0
L	U L a.b	0	0	0	0	0
DB/DI fully qualified[2)]	DB100.DBX10.3	0	0	0	0	0
DB/DI partly qualified	DBX10.3 with unknown DB number (e.g. after ON DB[MW20]	0.12	0.06	0.04	0.02	0.01
Timer		0	0	0	0	0
Counter		0	0	0	0	0
I/O access [1)]		1)	1)	1)	1)	1)

1) See Table for I/O direct access (see page 22)
2) The CPUs support symbolic programming. The fully qualified DB accesses (e.g. DB100.DBX 1.2) supported here generally cause no additional runtimes. This applies also for the ON DB command contained in the access.

Execution Times for Operand Access – Indirect Addressing

Operand Range	Example	Typical Execution Time in µs				
		312	314, 151	315, 154	317	319
Area-internal, register indirect addressing (AR1/AR2)	= A [AR1, P#1.1]	0.28	0.14	0.10	0.03	0.015
Area-crossing, register indirect addressing (AR1/AR2)	= [AR1, P#1.0]	0.88	0.44	0.33	0.11	0.05
Area indirect addressing	= A [MD2]	0.64	0.32	0.24	0.08	0.04
Addressing via parameter	U FC_Parameter	0.12	0.06	0.04	0.02	0.01
Access to FB instance data	U FC_Parameter, L Var_Stat	0.12	0.06	0.04	0.02	0.01
Timer	L T [MW2]	0.96	0.48	0.36	0.12	0.1
Counter	L Z [MW2]	0.96	0.48	0.36	0.12	0.1
I/O access 1)		1)	1)	1)	1)	1)

1) See Table for I/O direct access (see page 78)

Execution Times for Operand Access to I/0 – Direct and Indirect Addressing (PI / PO)

Operand	I/0 Areas	Example	Additional Execution Times for Operand Access in µs (typ.)							
			312	314	315-2 DP	315-2 PN/DP	317	319	151	154
Load Byte	Central	L PEB 0	14.3						67.8	
LoadWord	Central	L PEW 0	18.1						71.8	
Load DWord	Central	L PED 0	35.6						80.2	
Transfer Byte	Central	T PAB 0	11.2						63.4	
Transfer Word	Central	T PAW 0	12.7						67.4	
Transfer DWord	Central	T PAD 0	25.0						75.2	
Load Byte	Distributed (PB)	L PEB 0	–		3.9			1.7	3.9	
Load Word	Distributed (PB)	L PEW 0	–		4.1			1.8	4.1	
Load DWord	Distributed (PB)	L PED 0	–		4.2			1.8	4.2	
Transfer Byte	Distributed (PB)	T PAB 0	–		3.9			0.7	3.9	
Transfer Word	Distributed (PB)	T PAW 0	–		4.1			0.7	4.1	
Transfer DWord	Distributed (PB)	T PAD 0	–		4.3			0.8	4.3	
Load Byte	Distributed (PN)	L PEB 0	–			6.2		2.2	6.6	
Load Word	Distributed (PN)	L PEW 0	–			6.7		2.2	6.7	
Load DWord	Distributed (PN)	L PED 0	–			8.0		5.9	8.0	
Transfer Byte	Distributed (PN)	T PAB 0	–			7.8		2.2	7.8	
Transfer Word	Distributed (PN)	T PAW 0	–			7.9		2.2	7.9	
Transfer DWord	Distributed (PN)	T PAD 0	–			7.9		2.3	7.9	

Master Control Relay - active (MCR)

For the execution times in the active MCR area, an addition must be calculated for each command.

In the active MCR area, the execution time additions per command in µs are as follows:

CPU 312	CPU 314, IM 151-8	CPU 315, IM 154-8	CPU 317	CPU 319
0.4	0.3	0.2	0.07	0.04

Calculating the Execution Times for Area-Internal Memory-Direct Addressing

You will find a few examples here for calculating the execution times for the various methods of indirect addressing. Execution times are calculated for the CPU 315-2DP.

Calculating the Execution Times for Area-Internal Memory-Direct Addressing

Example: U M 0.0

Step 1: Execution time of the instruction (See Page 22 for times)

Instruction	Description	Typical Execution Time in µs
A	AND	**0.05**

Step 2: Execution times of operand access (See Page 74 for addressing)

Área de operando	Typical Execution Time in µs
M	**0**

Total execution time:

0.05 µσ
+ 0.00 µs
0.05 µσ

Calculating the Execution Times for Area-Internal Memory-Indirect Addressing

Example: U E [DBD 12]

Step 1: Execution time of the instruction (See Page 22 for times)

Instruction	Description	Typical Execution Time in µs
A	AND	**0.05**

Step 2: Execution times of operand access (See Page 75 for addressing)

Operand Range	Typical Execution Time in µs
Memory-Indirect Addressing	**0.24**

Total execution time:

0.05 µσ
\+ 0.24 µs
0.29 µσ

Calculating the Execution Time for Area-Internal Register-Indirect Addressing

Example: U E [AR1, P#34.3]

Step 1: Execution time of the instruction (See Page 22 for times)

Instruction	Description	Typical Execution Time in µs
A	AND	**0.05**

Step 2: Execution times of operand access (See Page 75 for addressing)

Operand Range	Typical Execution Time in µs
Area-Internal Register-Indirect Addressing	**0.10**

Total execution time:

0.05 µσ
\+ 0.10 µs
0.15 µσ

Calculating the Execution Time for Area-Crossing Register-Indirect Addressing

Example: U [AR1, P#23.1] ... mit P#E1.0 in AR1

Step 1: Execution time of the instruction (See Page 22 for times)

Instruction	Description	Typical Execution Time in µs
A	AND	**0.05**

Step 2: Execution times of operand access (See Page 75 for addressing)

Operand Range	Typical Execution Time in µs
Area-Crossing Register-Indirect Addressing	**0.33**

Total execution time:

0.05 µσ
\+ 0.33 µs
0.38 µσ

Execution Time for Addressing via Parameters

Example: A parameter ... with I 0.5 in the block parameter list

Step 1: Execution time of the instruction (See Page 22 for times)

Instruction	Description	Typical Execution Time in µs
A	AND	**0.05**

Step 2: Execution times of operand access (See Page 75 for addressing)

Operand Range	Typical Execution Time in µs
Addressing via parameter	**0.04**

Total execution time:

0.05 µσ
\+ 0.04 µs
0.09 µσ

Example of I/O Access

Example: L PEB 0 (centralized I/O)

Step 1: Time of the loading instructions - direct and indirect addressing (see page 22 for times)

Operation	Operand	Typical Execution Time in µs
L	B	**0.09**

Step 2: Execution times of operand access (see page 76 for addressing)

Operand	Additional Execution Times for Operand Access in µs
Load Byte	**13.7**

Total execution time:

00.09 µσ
+ 13.70 µs
13.79 µσ

Organisation Blocks (OB)

A user program for an S7-300 consists of blocks which contain the instructions, parameters, and data for the respective CPU. The individual CPUs of the S7-300 differ in the number of blocks which you can define for the respective CPU, and of those which are supplied by the operating system of the CPU. You can find a detailed description of the OBs and their use in the *STEP 7 online help system*.

Organisation Blocks	312	314, 151	315, 154	317	319	Starting Events (Hexadecimal Values)	
Cycle:							
OB 1	x	x	x	x	x	1101_H 1103_H	OB1 starting event Running OB1 start event (conclusion of the free cycle)
Time-of-day interrupt:							
OB 10	x	x	x	x	x	1111_H	Time-of-day interrupt event
Delay Interrupt:							
OB 20	x	x	x	x	x	1121_H	Delay interrupt event
OB 21	x	x	x	x	x	1122_H	Delay interrupt event
Cyclic interrupt:							
OB 32	x	x	x	x	x	1133_H	Cyclic interrupt event

Organisation Blocks	312	314	151	315, 154	317	319	Starting Events (Hexadecimal Values)	
OB 33	x	x	x	x	x	x	1134_H	Cyclic interrupt event
OB 34	x	x	x	x	x	x	1135_H	Cyclic interrupt event
OB 35	x	x	x	x	x	x [1]	1136_H	Cyclic interrupt event
Process interrupt:								
OB 40	x	x	x	x	x	x	1141_H	Process interrupt
DPV1-Interrupt (only DP-CPUs)								
OB 55	–	–	x	x	x	x	1155_H	Status interrupt
OB 56	–	–	x	x	x	x	1156_H	Update-interrupt
OB 57	–	–	x	x	x	x	1157_H	Manufacture-specific interrupt
Synchronous cycle interrupt								
OB 61 [2]	–	–	x	x	x	x	1164_H	Synchronous cycle interrupt

[1] Besides the ms-granular setting of the OB35 call interval, you can also select an µs-granular setting of the values for the OB35 in STEP 7 to be able to parameterize even the smallest cyclical interrupt of 500 µs and multiple (value range adjustable from 500 µs to 60000 ms).

[2] IM151-8 PN/DP CPU: synchronous cycle to PROFINET IO (not to PROFIBUS DP) CPU315, 154, 317 and 319: synchronous cycle either to PROFIBUS DP or to PROFINET IO (since only one synchronous cycle interrupt OB is available)

Organisation Blocks	312	314, 151	315, 154	317	319	Starting Events (Hexadecimal Values)	
Error responses:							
OB 80	x	x	x	x	x	3501_H 3502_H 3505_H 3507_H	Cycle time violation OB or FB request error Time-of-day interrupt elapsed due to time jump Multiple OB request error caused start info buffer overflow
Diagnostic interrupt:							
OB 82	x	x	x	x	x		3842_H Module o. k. 3942_H Module fault
OB 83	–	151[2)]	315[1)] 154[2)]	x[1)]	x [1)]	3854_H 3855_H 3961_H 3951_H 3961_H	PROFINET IO submodule inserted and corresponds with configured submodule PROFINET IO submodule inserted and does not correspond with unconfigured submodule Module inserted PROFINET IO module removed Module drawn

1) only CPU 315-2 PN/DP
2) for centralized peripherals and PROFINET IO

Organisation Blocks	312	314, 151	315, 154	317	319	Starting Events (Hexadecimal Values)	
OB 85	x	x	x	x	x	$35A1_H$	No OB or FB
						$35A3_H$	Error during access of a block by the operating system
						$39B1_H$	I/O access error during process image updating of the inputs (during each access)
						$39B2_H$	I/O access error during transfer of the process image to the output modules (during each access)
						$38B3_H$	I/O access error during process image updating of the inputs (outgoing event)
						$38B3_H$	I/O access error during process image updating of the inputs (incoming event)
						$38B4_H$	I/O access error during transfer of the process image to the output modules (outgoing event)
						$39B4_H$	I/O access error during transfer of the process image to the output modules (incoming event)

Organisation Blocks	312	314, 151	315, 154	317	319	Starting Events (Hexadecimal Values)	
OB 86	–	–	x	x	x	$32C9_H$	PROFIBUS DP: Station activated by SFC12 (mode 3)
						$33C9_H$	PROFIBUS DP: Station deactivated by SFC12 (mode 4)
						$38C4_H$	Distributed I/O: station failed, outgoing
						$39C4_H$	Distributed I/O: station failed, incoming
						$32CF_H$	PROFINET IO: Station activated by SFC12 (mode 3)
						$33CF_H$	PROFINET IO: Station deactivated by SFC12 (mode 4)
						38CBH	”PROFINET IO: Station return”
						39CBH	”PROFINET IO: Station failure
						38F8H	”PROFINET IO: Partial station return”
						39F8H	”PROFINET IO: Partial station failure”
OB 87	x	x	x	x	x	$35E1_H$	Incorrect frame identifier in GD $35E2_H$
						$35E2_H$	GD packet status cannot be entered in DB
						$35E6_H$	GD whole status cannot be entered in DB
Restart:							
OB 100	x	x	x	x	x	1381_H	Manual restart requests
						1382_H	Automatic restart requests

Organisation Blocks	312	314, 151	315, 154	317	319	Starting Events (Hexadecimal Values)	
Synchronous error responses:							
OB 121	x	x	x	x	x	2521_H	BCD conversion error
						2522_H	Range length error during reading
						2523_H	Range length error during writing
						2524_H	Range error during reading
						2525_H	Range error during writing
						2526_H	Timer number error
						2527_H	Counter number error
						2528_H	Alignment error during reading
						2529_H	Alignment error during writing
						2530_H	Write error during access to DB
						2531_H	Write error during access to DI
						2532_H	Block number error opening a DB
						2533_H	Block number error opening a DI
						2534_H	Block number error at FC call
						2535_H	Block number error at FB call
						$253A_H$	DB not loaded
						$253C_H$	FC not loaded
						$253E_H$	FB not loaded
OB 122	x	x	x	x	x	2944_H	I/O access error at nth read access (n > 1)
						2945_H	I/O access error at nth write access (n > 1)

Function Blocks (FB)

The following tables list the quantities, numbers, and maximal sizes of the function blocks, functions and data blocks that you can define in the individual CPUs of the S7-300.

Blocks	CPU 312	CPU 314, IM 151-8	CPU 315, IM 154-8	CPU 317	CPU 319
Quantity	1024	1024	1024	2048	4096
Admissible numbers	0 to 7999	0 to 7999	0 to 7999	0 to 7999	0 to 7999
Maximal size of an FB (process-relevant code)	32 kByte	64 kByte	64 kByte	64 kByte	64 kByte

Functions (FC)

Blocks	CPU 312	CPU 314, IM 151-8	CPU 315, IM 154-8	CPU 317	CPU 319
Quantity	1024	1024	1024	2048	4096
Admissible numbers	0 to 7999	0 to 7999	0 to 7999	0 to 7999	0 to 7999
Maximal size of an FC (process-relevant code)	32 kByte	64 kByte	64 kByte	64 kByte	64 kByte

Data Blocks

Blocks	CPU 312	CPU 314, IM 151-8	CPU 315, IM 154-8	CPU 317	CPU 319
Quantity	1024	1024	1024	2048	4096
Admissible numbers	1 to 16000	1 to 16000	1 to 16000	1 to 16000	1 to 16000
Maximal size of an FB (process-relevant code)	32 kByte	64 kByte	64 kByte	64 kByte	64 kByte

System Functions (SFC)

The following tables show the system functions offered by the

operating systems of the S7-300 CPUs and the execution times on the respective CPUs.

SFC No.	SFC Name	Description	Typical Execution Time in µs				
			312	314, 151	315, 154	317	319
0	SET_CLK	Sets the clock time	21	21	21	21	7
1	READ_CLK	Reads the clock time	7	7	7	6	3
2	SET_RTM	Sets the operating hours counter	6	6	6	5	3
3	CTRL_RTM	Starts/stops the operating hours counter	6	6	6	5	2
4	READ_RTM	Reads the operating hours counter	8	8	8	7	3
5	GADR_LGC	Determine logical channel address	26	26	26	18	12
6	RD_SINFO	Reads start information of the current OB.	11	11	11	5	3
7	DP_PRAL	Triggers a process interrupt from the user program of the CPU as DP slave through to DP master.	–	–	87	87	26
		concurrent running requests, max.	–	–	34 requests together with SFB 75 requests		
11	SYC_FR	Synchronizes groups of DP slaves	–	65 [1)]	65	54	23
		concurrent running requests, max.	–	–	2 requests		

[1)] supported only by IM151-8 PN/DP CPU (with DP master module inserted)

SFC No.	SFC Name	Description	Typical Execution Time in µs				
			312	314, 151	315, 154	317	319
12	D_ACT_DP	Activates or deactivates DP slaves	–	–	64	48	30
		concurrent running requests, max.	–	–	8 requests		
13	DPNRM_DG	Reads the DP-compliant slave diagnosis	–	–	33	23	10
		concurrent running requests, max.	–	–	4 requests		
14	DPRD_DAT	Reads/writes consistent data (n bytes)	–	–	27	20	15
15	DPWR_DAT	Reads/writes consistent data (n bytes)	–	–	26	24	15
17	ALARM_SQ	Generates block-related messages that can be acknowledged	126	126	126	99	67
18	ALARM_S	Generates block-related messages that can not be acknowledged	126	126	126	101	68
19	ALARM_SC	Acknowledgment state of the last ALARM_SQ received message	27	27	27	20	5
20	BLKMOV	Copies variables within the working memory	10 + 0.01/ Byte	10 + 0.01/ Byte	10 + 0.01/ Byte	7 + 0.01/ Byte	2 + 0.003 / Byte
21	FILL	Sets array default variables within the working memory	10 + 0.035 / Byte	10 + 0.035 / Byte	10 + 0.035 / Byte	6 + 0.035/ Byte	3 + 0.01 / Byte
22	CREAT_DB	Generates a data block	86	86	86	63	50

1) supported only by IM151-8 PN/DP CPU (with DP master module inserted)

SFC No.	SFC Name	Description	Typical Execution Time in µs				
			312	314, 151	315, 154	317	319
23 [1]	DEL_DB	Deletes a data block	94	94	94	87	52
		concurrent running requests, max.	21 requests				
24	TEST_DB	Tests a data block	13	13	13	7	5
28	SET_TINT	Sets the times of a time-of-day interrupt	17	17	17	11	5
29	CAN_TINT	Cancels a time-of-day interrupt	8	8	8	4	2
30	ACT_TINT	Activates a time-of-day interrupt	10	10	10	5	2
31	QRY_TINT	Queries the status of a time–of-day interrupt	11	11	11	6	2
32	SRT_DINT	Starts a delay interrupt	10	10	10	7	7
33	CAN_DINT	Cancels a delay interrupt	10	10	10	5	5
34	QRY_DINT	Queries started delay interrupts	8	8	8	3	3
36	MSK_FLT	Masks sync faults	8	8	8	5	3
37	DMSK_FLT	Enables sync faults	8	8	8	5	3
38	READ_ERR	Reads event status register	7	7	7	5	2

[1] The SFC23 deletes data blocks in the operating mode RUN. If a SFC 23 call is present in the loaded project then additional tests are carried when the data blocks are accessed. This can increase the command run time on the operand area DB. If a data block is accessed that was deleted in RUN by SFC 23 then the programming error OB (OB 121) is called. DBs are deleted in the background and the process may take as long as the OB1 cycle. Freeing up memory resources may claim many OB1 cycles.

SFC No.	SFC Name	Description	Typical Execution Time in µs				
			312	314, 151	315, 154	317	319
39	DIS_IRT	Disables the handling of new interrupts	24	24	24	15	9
40	EN_IRT	Enables the handling of new interrupt events	23	23	23	20	13
41	DIS_AIRT	Delays the handling of interrupts	24	24	24	24	10
42	EN_AIRT	Enables the handling of interrupts	13	13	13	13	7
43	RE_TRIGR	Re-triggers the scan time monitor	21	21	21	13	12
44	REPL_VAL	Copies a substitute value into accumulator 1	5	5	5	4	3
46	STP	Forces the CPU into the STOP mode	no numerical data				
47	WAIT	Delays program execution in addition to waiting times	Waiting time + 0.1 % of this				
49	LGC_GADR	Converts a free address to the slot and rack for a module	20	20	20	10	8

SFCNo.	SFC Name	Description	Typical Execution Time in µs				
			312	314, 151	315, 154	317	319
50	RD_LGADR	Reads all the declared free addresses for a module	38	38	38	22	18
51	RDSYSST	Reads out the information from the system state list. SFC 51 is not interruptible through interrupts.	9 + 0.1 / Byte	9 + 0.1 / Byte	9 + 0.1 / Byte	7 + 0.1 / Byte	3 + 0.1 / Byte
		concurrent running requests, max.	4 requests				
52	WR_USMSG	Writes specific diagnostic information in the diagnostic buffer	290	290	290	290	60
55	WR_PARM	Writes dynamic parameters to a module	190	190	190	190	190
		concurrent running requests, max.	1 request				
56	WR_DPARM	Writes predefined dynamic parameters to a module	95	95	95	95	95
		concurrent running requests, max.	1 request				

SFC No.	SFC Name	Description	Typical Execution Time in µs				
			312	314, 151	315, 154	317	319
57	PARM_MOD	Assigns a module's parameters	95	95	95	95	95
		concurrent running requests, max	1 request				
58	WR_REC	Writes a module-specific data record	388 + 10 / Byte				350 + 10 / Byte
		concurrent running requests to different modules to different modules, max	4 requests together with SFB 53 requests			8 requests together with SFB 53 requests	
59	RD_REC	Reads a module-specific data record	461 + 12 / Byte				432 + 12 / Byte
		concurrent running requests to different modules, max.	4 requests together with SFB 52 requests			8 requests together with SFB 52 requests	
64	TIME_TICK	Reads out the system time	6	6	6	4	2

SFC No.	SFC Name	Description	Typical Execution Time in µs				
			312	314, 151	315, 154	317	319
65	X_SEND [1]	Sends data to a communication partner external to your own S7 station	15	15	15	13	8
		The maximum number of simultaneous SFC65, SFC67, SFC68, SFC72 or SFC73 jobs to different remote communication partners (Note: only one SFC65, SFC67, SFC68, SFC72 or SFC73 job at a time is possible to a remote communication partner).	4 requests	10 requests	14 requests	30 requests	
66	X_RCV [1]	Receives data from a communication partner external to your own S7 station	19	19	19	9	8
		The maximum number of simultaneous SFC65, SFC67, SFC68, SFC72 or SFC73 jobs to different remote communication partners (Note: only one SFC65, SFC67, SFC68, SFC72 or SFC73 job at a time is possible to a remote communication partner).	4 requests	10 requests	14 requests	30 requests	

[1] Does not apply to IM151-8 PN/DP CPU

SFC No.	SFC Name	Description	Typical Execution Time in µs				
			312	314, 151	315, 154	317	319
67	X_GET [1]	Reads data from a communication partner external to your own S7 station	18	18	18	12	5
		The maximum number of simultaneous SFC65, SFC67, SFC68, SFC72 or SFC73 jobs to different remote communication partners (Note: only one SFC65, SFC67, SFC68, SFC72 or SFC73 job at a time is possible to a remote communication partner).	4 requests	10 requests	14 requests	30 requests	30 requests
68	X_PUT [1]	Writes data to a communication partner external to your own S7 station	18	18	18	12	5
		The maximum number of simultaneous SFC65, SFC67, SFC68, SFC72 or SFC73 jobs to different remote communication partners (Note: only one SFC65, SFC67, SFC68, SFC72 or SFC73 job at a time is possible to a remote communication partner).	4 requests	10 requests	14 requests	30 requests	30 requests
69	X_ABORT [1]	Aborts connection to a communication partner external to your own S7 station	7	7	7	5	5

[1] Does not apply to IM151-8 PN/DP CPU

SFC No.	SFC Name	Description	Typical Execution Time in µs				
			312	314, 151	315, 154	317	319
70	GEO_LOG	Determine module start address	23	23	23	9	8
71	LOG_GEO	Querying the module slot belonging to a logical address	21	21	21	11	8
72	I_GET	Reads data from a communication partner within your own S7 station	36	36	36	28	15
		The maximum number of simultaneous SFC65, SFC67, SFC68, SFC72 or SFC73 jobs to different remote communication partners (Note: only one SFC65, SFC67, SFC68, SFC72 or SFC73 job at a time is possible to a remote communication partner).	4 requests	10 requests	14 requests	30 requests	30 requests

SFC No.	SFC Name	Description	Typical Execution Time in µs				
			312	314, 151	315, 154	317	319
73	I_PUT	Writes data to a communication partner within your own S7 station	28	28	28	28	15
		The maximum number of simultaneous SFC65, SFC67, SFC68, SFC72 or SFC73 jobs to different remote communication partners (Note: only one SFC65, SFC67, SFC68, SFC72 or SFC73 job at a time is possible to a remote communication partner).	4 requests	10 requests	14 requests	30 requests	30 requests
74	I_ABORT	Aborts connection to a communication partner within your own S7 station	8	8	8	6	2
81	UBLKMOV	Copy the variable without interruption, length of the data to be copied up to 512 bytes	11 + 0.01 / Byte	11 + 0.01 / Byte	11 + 0.01 / Byte	8 + 0.01 / Byte	3
82	CREA_DBL	Create data block in load memory.	46	46	46	39	20
		concurrent running requests, max.	3 requests				
83	READ_DBL	Read from a data block in load memory	47	47	47	36	20
		concurrent running requests, max.	3 requests				

SFC No.	SFC Name	Description	Typical Execution Time in µs				
			312	314, 151	315, 154	317	319
84	WRIT_DBL	Write to a data block in load memory.	50	50	50	36	20
		concurrent running requests, max.	3 requests				
99[1]	WWW	Connection between user program and web server	-	17	17	15	4
101	RTM	Handling the Run-time meter	8	8	8	7	3
102	RD_DPARA	Read predefined parameter.	62	62	62	53	30
		concurrent running requests, max.	1 request				
103	DP_TOPOL	Detemine bus topology in a DP Master system fist call	_	25[2]	25	7	7
105	READ_SI	Read dynamically assigned system resources	47 + 0.61 per alarm	47 + 0.61 per alarm	47 + 0.61 per alarm	47 + 0.26 per alarm	15 + 0.1 per alarm
106	DEL_SI	Enable dynamically assigned system resources	146 + 3.8 per alarm	146 + 3.8 per alarm	146 + 3.8 per alarm	140 + 3.6 per alarm	107 + 3.6 per alarm
107	ALARM_DQ	Acknowledgeable block-related messages create first call	127	127	127	98	69

1) Only for PROFINET CPUs (CPU 31x-2 PN/DP, CPU 319-3 PN/DP, IM 15x-8 PN/DP CPU)
2) supported only by IM151-8 PN/DP CPU (with DP master module inserted)

SFC No.	SFC Name	Description	Typical Execution Time in µs				
			312	314, 151	315, 154	317	319
108	ALARM_D	Not acknowledgeable block-related messages create first call	129	129	129	99	69
109	PROTECT	Activate write protection 2)	4	4	4	3	2
112	PN_IN 1)	Update inputs of the user program interface of the PROFINET components	–	778	760	612	197
113	PN_OUT 1)	Update outputs of the user program interface of the PROFINET components	–	604	608	446	158
114	PN_DP 1)	Update DP interconnection	–	153	150	132	105
126	SYNC_PI	Update the process image partition of the inputs in a synchronous cycle	–	30 + 0.2 / Byte 3)	30 + 0.2 / Byte	29 + 0.2 / Byte	22 + 0.15 / Byte
		concurrent running requests, max.	1 request				
127	SYNC_PO	Update the process image partition of the outputs in a synchronous cycle	–	29 + 0.2 / Byte 3)	29 + 0.2 / Byte	28 + 0.2 / Byte	25 + 0.15 / Byte
		concurrent running requests, max.	1 request				

1) for CPU 315-2 PN/DP, CPU 317-2 PN/DP: The runtimes of these blocks depend on their respective interconnection configuration. See also manual CPU 31xC and CPU 31x, technical data: chapter: “Cycle and response times, extending the OB1 cycle for cyclical PROFINET interconnections”.

2) It is recommended to protect the CPU with a password to prevent unauthorized access. Please note the characteristic features for fail-safe systems.

3) supported only by IM151-8 PN/DP CPU (with DP master module inserted)

System Function Blocks (SFB)

The following table lists the system function blocks supplied by the operating system of the S7-300's CPUs, and the execution times on the respective CPUs.

SFB No.	SFB Name	Description	Typical Execution Time in µs				
			312	314, 151	315, 154	317	319
0	CTU	Counts up	13	13	13	9	4
1	CTD	Counts down	11	11	11	8	3
2	CTUD	Counts up and counts down	11	11	11	9	3
3	TP	Generates a pulse	13	13	13	11	5
4	TON	Delays a leading edge	13	13	13	9	5
5	TOF	Delays a falling edge	12	12	12	8	3
32	DRUM	Implements a sequence processor with a maximum of 16 s	40	40	40	20	10

SFB No.	SFB Name	Description	Typical Execution Time in µs				
			312	314, 151	315, 154	317	319
52	RDREC	Read Data set from DP slave, PROFINET IO-Device or central module	483 + 12 / Byte	483 + 12 / Byte	483 + 12 / Byte	469 + 12 / Byte	432 + 12 / Byte
		concurrent running requests to different modules, max.	4 requests together with SFC 59 requests			8 requests together with SFC 59 requests	
53	WRREC	Write Data set to DP slave, PROFINET IO-Device or central module	429 + 10 / Byte				350 + 10 / Byte
		concurrent running requests to different modules, max.	4 requests together with SFC 58 requests			8 requests together with SFC 58 requests	
54	RALRM	Read out interrupt status information from interrupts of a DP slave, PROFINET IO-Device or of a central module in the respective OB	31	31	31	27	7
		concurrent running requests, max.	1 request				

SFB No.	SFB Name	Description	Typical Execution Time in µs				
			312	314, 151	315, 154	317	319
73	RCVREC	Receive data records in an I-device from a higher-level IO controller	–	90 + 0,015 / Byte[2)]	90 + 0,015 / Byte	60 + 0,01 / Byte	35 + 0,005 / Byte
75	PRVREC	Provide data records in an I-device to a higher-level IO controller	–	90 + 0,015 / Byte[2)]	90 + 0,015 / Byte	60 + 0,01 / Byte	35 + 0,005 / Byte
75	SALRM	Set desired interrupts of I-slaves	–	–	41	32	30
		concurrent running requests, max.	–	–	34 requests together with SFC 7 requests		
81	RD_DPAR	Reading predefined parameters	50	50	50	30	20
		concurrent running requests, max.	4 requests				
104[1)]	IP_CONF		–	84	41	26	15

1) Only for PROFINET CPUs (CPU 31x-2 PN/DP, CPU 319-3 PN/DP, IM 15x-8 PN/DP CPU)
2) supported only by IM151-8 PN/DP CPU

Standard Function Blocks for S7-Communication via CP

For some communication services, pre-fabricated blocks are available as an interface your STEP7 user program. See also Standard-Library, Communication Blocks.

FB No.	FB Name	Description	may be used with	
			CPU 312, 314, 315-2 DP	CPU 315-2 PN/DP, 317-2 PN/DP, 319-3 /PN/DP, IM 151-8, IM 154-8
8	USEND	Uncoordinated data sending	Communication via CP	Communication via CP or integrated PROFINET Interface[1)]
9	URCV	Uncoordinated data reception		
12	BSEND	Block-oriented data sending		
13	BRCV	Block-oriented data reception		
14	GET	Read data from a remote CPU		
15	PUT	Write data from a remote CPU		
28*	USEND_E	Uncoordinated sending of data with expanded sending ranges SD_1 to SD_4	-	Communication via integrated PROFINET interface
29*	URCV_E	Uncoordinated reception of data with expanded reception ranges RD_1 to RD_4	-	
34*	GET_E	Reading remote CPU with expanded reception ranges RD_1 to RD_4	-	
35*	PUT_E	Writing data to a remote CPU (with expanded ranges SD_1 to SD_4 to be written)	-	

1) for IM 151-8 and IM 154-8 only via the integrated PROFINET interface
*) ab V3.2

FC No.	FC-Name	Description	may be used with	
			CPU 312, 314, 315-2 DP	CPU 315-2 PN/DP, 317-2 PN/DP, 319-3 /PN/DP, IM 151-8, IM 154-8
62	C_CNTRL	Request connection status which belongs to a local connection.	Communications via CP	Communication via CP or integrated PROFINET Interface

Function Blocks for open system interconnection over Industrial Ethernet

In order to be able to exchange data via user programs with other TCP/IP-capable communication partners, STEP7 places FBs and UDTs at your disposal. These blocks are saved in the Standard-Library,Communication Blocks.

FB No.	FB Name	Description	CPU 315-2 PN/DP CPU 317-2PN/DP CPU 319-3 PN/DP IM 151-8 PN/DP CPU IM 154-8 PN/DP CPU	Communications protocol
62	TSEND	Sending of data	with firmware as of V3.2	TCP, ISO-on-TCP
64	TRCV	Receiving of data		TCP, ISO-on-TCP
65	TCON	Establishing a communication link		TCP, ISO-on-TCP, UDP
66	TDISCON	Disconnecting a communication link		TCP, ISO-on-TCP, UDP
67	TUSEND	Sending of data		UDP
68	TURCV	Receiving of data		UDP

IEC Functions

You can use the following functions in STEP 7:

These blocks are saved in the Standard Library, IEC Function-Blocks in STEP 7.

FC No.	FC Name	Description
DATE_AND_TIME		
3	D_TOD_DT	Concatenates the data formats DATE and TIME_OF_DAY (TOD) and converts to data format DATE_AND_TIME.
6	DT_DATE	Extracts the DATE data format from the DATE_AND_TIME data format.
7	DT_DAY	Extracts the day of the week from the data format DATE_AND_TIME.
8	DT_TOD	Extracts the TIME_OF_DAY data format from the DATE_AND_TIME data format.
Time Formats		
33	S5TI_TIM	Converts S5 TIME data format to TIME data format
40	TIM_S5TI	Converts TIME data format to S5 TIME data format
Duration		
1	AD_DT_TM	Adds a duration in the TIME format to a time in the DT format. The result is a new time in the DT format.
35	SB_DT_TM	Subtracts a duration in the TIME format from a time in the DT format. The result is a new time in the DT format.
34	SB_DT_DT	Subtracts two times in the DT format. The result is a duration in the TIME format.

FC No.	FC Name	Description
Compare DATE_AND_TIME		
9	EQ_DT	Compares the contents of two variables in the DATE_AND_TIME format for equal to.
12	GE_DT	Compares the contents of two variables in the DATE_AND_TIME format for greater than or equal to.
14	GT_DT	Compares the contents of two variables in the DATE_AND_TIME format for greater than.
18	LE_DT	Compares the contents of two variables in the DATE_AND_TIME format for less than or equal to.
23	LT_DT	Compares the contents of two variables in the DATE_AND_TIME format for less than.
28	NE_DT	Compares the contents of two variables in the DATE_AND_TIME format for not equal to.
Compare STRING		
10	EQ_STRNG	Compares the contents of two variables in the STRING format for equal to.
13	GE_STRNG	Compares the contents of two variables in the STRING format for greater than or equal to.
15	GT_STRNG	Compares the contents of two variables in the STRING format for greater than.
19	LE_STRNG	Compares the contents of two variables in the STRING format for less than or equal to.
24	LT_STRNG	Compares the contents of two variables in the STRING format for less than.
29	NE_STRNG	Compares the contents of two variables in the STRING format for not equal to.

FC-Nr.	FC-Name	Description
STRING Variable Processing		
21	LEN	Reads the length of a STRING variable.
20	LEFT	Reads the first L characters of a STRING variable.
32	RIGHT	Reads the last L characters of a STRING variable.
26	MID	Reads the middle L characters of a STRING variable (starting at the defined character).
2	CONCAT	Concatenates two STRING variables in one STRING variable.
17	INSERT	Inserts a STRING variable into another STRING variable at a defined point.
4	DELETE	Deletes L characters of a STRING variable.
31	REPLACE	Replaces L characters of a STRING variable with a second STRING variable.
11	FIND	Finds the position of the second STRING variable in the first STRING variable.

FC No.	FC Name	Description
Format Conversions with STRING		
16	I_STRNG	Converts a variable from INTEGER format to STRING format.
5	DI_STRNG	Converts a variable from INTEGER (32-bit) format to STRING format.
30	R_STRNG	Converts a variable from REAL format to STRING format.
38	STRNG_I	Converts a variable from STRING format to INTEGER format.
37	STRNG_DI	Converts a variable from STRING format to INTEGER (32-bit) format.
39	STRNG_R	Converts a variable from STRING format to REAL format.
Number Processing		
22	LIMIT	Limits a number to a defined limit value.
25	MAX	Selects the largest of three numeric variables.
27	MIN	Selects the smallest of three numeric variables.
36	SEL	Selects one of two variables.

see also STEP 7 Online Help

System Status Sublist

SSL-ID	Index	Information Function
		Module identification
0111_H		Identification data record corresponding to the specified index
	0001_H	CPU type and version number
	0006_H	Identification of the basic hardware
	0007_H	Identification of the basic firmware
		CPU features
0012_H	-	All features
0112_H		Features of a group
	0000_H	STEP 7 processing
	0100_H	Time system in the CPU
	0200_H	System behavior of the CPU
	0300_H	STEP 7 instruction set
$0F12_H$	-	Header information only
		User memory areas
0013_H	-	All data records of available user memory areas
0113_H		One data record for the specified memory area
	0001_H	Work memory

SSL-ID	Index	Information Function
		System area
0014_H	-	Data records of all system areas
$0F14_H$	-	Header information only
		Block types
0015_H	-	Data records of all block types
		Status of module LEDs
0019_H	-	Read all LED statuses
$0F19_H$	-	Header information only

SSL-ID	Index	Information Function
		Component identification
$001C_H$	-	Read all data records
$011C_H$		Data record to specified index
	0001_H	Station name
	0002_H	Module name
	0003_H	Higher level designation of the module
	0004_H	Copyright entry
	0005_H	Serial number of the module
	0007_H	Module type name
	0008_H	Serial number of the micro memory card
	0009_H	Manufacturer and profile of a CPU module
	$000A_H$	OEM identifier
	$000B_H$	Location designation
$01FC_H$	-	Header information only

SSL-ID	Index	Information Function
		Interrupt status
0222_H		Data record to specified interrupt
	OB No.	Number of the OB (Only OB1)
		Assignment between partial process images and CPUs (only for CPU315-2DP, CPU 315-2 PN/DP, CPU 317-2 PN/DP))
0025_H	-	Assignment between all partial process images and OBs
0125_H	PPI No. (no. of the partial process image)	Assignment between a partial process image and the corresponding OB
0225_H	OB No.	Assignment between an OB and the corresponding partial process images
$0F25_H$	-	Only SZL sublist information
		Communication status data
0132_H		Communication status information to the specified communication unit (only one data record)
	0004_H	OMS/ contactor
	0005_H	Diagnostics
	0008_H	Time system (TIME)
	$000B_H$	Runtime meter (32-bit) 0 to 7
	$000C_H$	Runtime meter (32-bit) 8 to 15
0232_H		Communications status information on specified communication unit

SSL-ID	Index	Information Function
	0004_H	OMS/ contactor

SSL-ID	Index	Information Function
		Status of module LEDs
0074_H	-	Read all LED statuses
0174_H		Read individual LED statuses
	0001_H	GE, group error
	0004_H	RUN, RUN-LED
	0005_H	STOP, STOP-LED
	0006_H	FRCE, Force-LED
	$000B_H$	BUS1F-LED
	$000C_H$	BUS2F-LED
	0014_H	BUS3F-LED
	0015_H	MAINT-LED
		Module status information
0591_H	-	Module status information of all submodules that know a host
$0A91_H$	-	Module status information of all DP master systems known to CPU (only CPUs with DP interface)
$0C91_H$		Module status information of a module
	Any logical address of a module/submodule	Module status information of a module on logical address

SSL-ID	Index	Information Function
$0D91_H$		Module status information of a rack or station
	Centralized configuration: 0000h: Rack 0 0001h: Rack 1 0002h: Rack 2 0003h: Rack 3 **PROFIBUS DP:** xxyyh: DP-Subnetz-ID/Station No. **PROFINET IO:** Module location address of the PROFINET IO device: Bit 15: is always = 1 Bit 11-14: PN IO Subsystem-ID (value range 100-115; in which 0-15 must only be specified) Bit 0-10: Station number of the PROFINET IO Device	Module status information of all modules in specified rack/station

SSL-ID	Index	Information Function
		Rack/station status information
0092_H		Setpoint status of racks in centralized configuration or of stations in a subnet
	0000_H	Information on the status of a rack in the centralized configuration
	DP master system ID	Information on the status of stations in subnet
0292_H		Actual status of racks in centralized configuration or of stations in a subnet
	0000_H	Information on the status of a rack in the centralized configuration
	DP master system ID	Information on the status of stations in subnet
0692_H		Diagnostics status of racks in centralized configuration or of stations in a subnet
	0000_H	Information on the status of a rack in the centralized configuration
	DP master system ID	Information on the status of stations in subnet

SSL-ID	Index	Information Function
		Rack/station status information
0094_H		Setpoint status of a rack in centralized configuration or of stations in a subnet
	0000_H	Information on the status of racks in centralized configuration
	DP master system ID or PN IO Subsystem No.	Information on the status of stations in subnet
0194_H		Activation status of stations of a subnet (only CPU with DP and/ or PN interface)
	DP master system ID or PN IO Subsystem No.	Information on the status of stations in subnet
0294_H		Actual status of racks in centralized configuration or of stations in a subnet
	0000_H	Information on the status of racks in the centralized configuration
	DP master system ID or PN IO Subsystem No.	Information on the status of stations in subnet
0694_H		Diagnostics status of racks in centralized configuration or of stations in a subnet
	0000_H	Information on the status of racks in the centralized configuration
	DP master system ID or PN IO Subsystem No.	Information on the status of stations in subnet
$0F94_H$	-	Header information only

SSL-ID	Index	Information Function
		Extended DP master system information
0195_H	xxyyh: DP master system ID/00h	Extended DP master system information of a DP master system (only CPU 315-2 DP, CPU 315-2 PN/DP, CPU 317-2 PN/DP)
$0F95_H$	-	Header information only (only CPU 315-2 DP, CPU 315-2 PN/DP, CPU 317-2 PN/DP)
		Submodule status information
0696_H	Any logical address of a module/submodule	Status data of all submodules of a module
$0C96_H$	Any logical address of a module/submodule	Status data of a submodule
		Diagnostic buffer
$00A0_H$		All input event information (in the RUN of CPU default mode outputs only 10 entries; the number of event infos output in RUN can be parameterized from 10 – 499)
$01A0_H$	x	The ”x” most recent input event infos
$0FA0_H$	-	Header info SZL only

SSL-ID	Index	Information Function
		Diagnostics data of modules
$00B1_H$	Any logical address of a module/submodule	The first four diagnostic bytes of a module (diagnostics data record DS0)
$00B2_H$	Rack and slot number	All diagnostics data of a module (diagnostics data record DS1 – only for centrally mounted modules)
$00B3_H$	Any logical address of a module/submodule	All diagnostics data of a module (diagnostics data record DS1)
$00B4_H$	Logical start address (diagnostics address of the slave)	Standard diagnostics data of a DP slave (only CPUs with DP interface)

Alphabetical Index of Instructions